NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DEVELOPMENT OF A QUALITY MANAGEMENT METRIC (QMM) MEASURING SOFTWARE PROGRAM MANAGEMENT QUALITY

by
Martin J. Machniak

December 1999

Thesis Co-Advisor:	John Osmundson
Thesis Co-Advisor:	J. Bret Michael

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 1999	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE **DEVELOPMENT OF A QUALITY MANAGEMENT METRIC (QMM) MEASURING SOFTWARE PROGRAM MANAGEMENT QUALITY**	**5. FUNDING NUMBERS**
6. AUTHOR(S) Machniak, Martin J.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT

The quality of software management in a development program is a major factor in determining the success of a program. The four main areas where a software program manager can affect the outcome of a program are requirements management, estimation/planning management, people management, and risk management. By using current researched practices, interviews with senior program managers, and focus group data, the thesis examines the four areas for practices and structure that a software program manager may implement to positively affect the program. The thesis develops a Quality Management Metric (QMM) to measure the performance of the software manager. The QMM score is determined via a survey consisting of a two-part questionnaire for each of the four main areas examined. The thesis evaluated three software programs for a QMM score. Informal verification and validation of the metric compared the QMM percentile score to an overall program success score and yielded positive correlation. The establishment of this methodology to quantify the quality of software management is an important step in evaluation of how past and current programs are managed and can serve as a template to improve software management performance in the future.

14. SUBJECT TERMS Software Management, Requirements Management, Estimation/Planning Management, Risk Management, People Management, Quality Management Metric (QMM)	**15. NUMBER OF PAGES** 146
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

DEVELOPMENT OF A QUALITY MANAGEMENT METRIC (QMM) MEASURING SOFTWARE PROGRAM MANAGEMENT QUALITY

Martin J. Machniak
B.S./E.E.C.S., University of California, Berkeley, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1999

Author: ______________________
Martin J. Machniak

Approved by: James Bret Michael
J. Bret Michael, Thesis Co-Advisor

John S. Osmundson
John Osmundson, Thesis Co-Advisor

____
Luqi, Chairman
Software Engineering

ABSTRACT

The quality of software management in a development program is a major factor in determining the success of a program. The four main areas where a software program manager can affect the outcome of a program are requirements management, estimation/planning management, people management, and risk management. By using current researched practices, interviews with senior program managers, and focus group data, the thesis examines the four areas for practices and structure that a software program manager may implement to positively affect the program. The thesis develops a Quality Management Metric (QMM) to measure the performance of the software manager. The QMM score is determined via a survey consisting of a two-part questionnaire for each of the four main areas examined. The thesis evaluated three software programs for a QMM score. Informal verification and validation of the metric compared the QMM percentile score to an overall program success score for the entire program and yielded positive correlation. The establishment of this methodology to quantify the quality of software management is an important step in evaluation of how past and current programs are managed and can serve as a template to improve software management performance in the future.

TABLE OF CONTENTS

ACKNOWLEGEMENT

Sincere appreciation to
Dr. John Osmundson and Dr. J. Bret Michael
For their guidance and support in this endeavor
And also to
All interviewees, survey, and focus group participants
Especially,
Gerry Nifontoff and L. Preston Brooks, Jr.

I. INTRODUCTION AND BACKGROUND

A. MOTIVATION

Software metrics is the buzzword today in both development and maintenance activities. Process metrics focus on the activities involved in software development or maintenance. The product metric focuses on individual aspects of the item (usually volume) under development or maintenance. Both metrics typify most program performance evaluations and ignore any consideration of the quality of software management [Ref. 1]. These evaluations assume all software management is the same and doesn't detract from, or add to, conclusions derived from the other metrics. In 1981, Barry Boehm [Ref. 1] wrote,

> Poor management can increase software costs more rapidly than any other factor.

On this basis, software management can be considered the third leg of what is referred to as the golden triangle of software metrics.

B. SOFTWARE MANAGEMENT COMPONENTS AND GOALS

Software management quality comes in a wide variety of forms, but most deal with four distinct areas: requirements, estimation/planning, personnel and risk management. Current papers on software management are very subjective. Anecdotal stories are detailed, case studies are outlined, and bits and pieces of good advice are presented [Ref. 2]. Emphasis is placed in pointing out problems instead of solutions, and very little in providing objective indicators to measure management quality. This thesis builds an objective, repeatable metric to determine quality management, measure improvement, and predict future success levels of projects.

The goal is to determine a structured set of inquiries to quantitatively measure software management quality. The inquiries are organized into a questionnaire and minimize open-ended subjective essay-type answers. The questions are designed to confine responses, with the answer to be correlated to a standardized measure. Three software programs will be examined for establishment of these criteria. The three software programs are the Surveillance Towed Array Sensor System (SURTASS) 1989, the Financial Information Support System/Expenditure Tracking System (FISS/ETS) 1998, and the Tactical Environmental Support System/ Naval Integrated Tactical Environmental System (TESS/NITES) 1999. These programs are cross sections of typical Department of Defense software development and maintenance efforts, and serve to illustrate varied software management practices. In order to encourage complete openness with the survey, the results of the surveys will not identify which of the three programs they refer to. Instead the three programs will be randomly referred to as program A, program B, and program C.

Collectively, measures in the following four areas will give an objective view on the quality of the software management. Thus, two programs scoring equally on product and process metrics can be further measured and compared on the basis of the quality of their management. This provides a more comprehensive look at a software program.

1. Requirements Management

Requirements management focuses on managing the process of extracting, developing, defining, and refining the requirements of a software program [Ref. 3]. It is not the intent of this thesis to develop a product or process metric for requirements. Multitudes of product and process metrics exist in this area [Ref. 4]. Alan M. Davis and Dean A. Leffingwell [Ref. 5] state that,

> Requirements are capabilities and objectives to which software must conform and are the common thread for all development (and maintenance) activities. Requirements management is the process of eliciting, documenting, organizing, and tracking changing requirements and communicating this information across the project team. Implementing (quality) requirements management ensures that iterative and unanticipated changes are maintained throughout the project lifecycle.

Quality management of a program's requirements must establish procedures and structure to ensure that requirements specifications are complete, consistent, readable, lack ambiguity, can be traced to origins, and do not arbitrarily contain design stipulation [Ref. 5]. Each requirement should be a singular idea [Ref. 3]. Good management addresses the requirement attributes. These include managing customer benefit, the requirements author and/or responsible parties, the corresponding effort, the development priority, rationale, and relationships to other requirements. The effort in tracking status, dates, and versions also is a determinate of quality management [Ref. 5].

2. Estimation/Planning Management

Estimations are the basis of which planning is performed on a program [Ref. 6]. The estimation/planning management section does not seek to choose or purport a specific estimation technique. This area seeks to quantify the management effort of the estimation process. The questions are if the choice of a technique is appropriate and how well that technique is implemented.

3. People Management

The people management section encompasses not only such issues as the program manager's ability to allocate human resources appropriately and ensure an appropriate working environment, it also includes communication and leadership. This includes not only the communication and leadership skills of the program manager, but also the structure set up for communication and mentoring leadership for the entire program. This thesis looks at management of people from a specific software development/management perspective. It examines such questions as, does management create the proper environment through good working conditions and an appropriate reward structure, and does management create unnecessary overlaps or underlaps through poor organization, delegation, and task monitoring. This section is an exclusive focus on the unique qualities and needs of people working in a software development environment.

4. Risk Management

An overarching theme that runs through each of these sections is risk management. Ultimately, it is management's ability to identify and resolve high-risk elements early that will have the greatest impact on the success or failure of a software program [Ref. 7]. It is difficult to objectively measure subjective decisions regarding risk management. It then elevates the priority to objectively measure the effort and structure a program has dedicated to risk assessments.

II. REQUIREMENTS MANAGEMENT

A. COMPONENTS AND CRITERIA

In developing software, requirements are the reason why it is done in the first place. Without requirements, there is no need for development. [Ref. 8]

A software development project generally consists of initial requirements, refined requirements, implementation of requirements, and then testing of the product for conformance to the requirements [Ref. 9]. A software maintenance or follow-on upgrade development deals with new and modified requirements. A well-documented requirement is a single idea or function [Ref. 3]. The requirement is easy to understand and is testable in some fashion [Ref. 3]. For these reasons, the management of requirements is an important measure of the quality of program management. For instance, can the program manager control the process of development, prioritization, and implementation of requirements, given constraints in any of these areas? Constraints can be in the form of mandates to employ a certain development process, a selected architecture, or by a predetermined set of requirements.

The program manager must identify and ensure that all stakeholders are involved in the initial requirements list development. Failure to include all parties at the start will most likely spell trouble down the line [Ref. 10]. Steve McConnell [Ref. 11], in his IEEE Software article listing Software's Ten Essentials, calls the product specification, the software program's compass. He states,

> ...without one, you can perform the work of Hercules and still not produce a working product because the work in aggregation hasn't been aimed in any particular direction. Without good direction, any individual's work can go the wrong direction and different people can work at cross-purposes.

Most program managers regard requirements as the contract between the developer and the customer on a program [Ref. 12]. The program manager manages customer's expectations by managing the requirements [Ref. 12]. Generally, a program is created to fill a user or customer need. In the Department of the Navy, that could mean the fleet has a need for some capability. That capability may be translated into a submarine tracking software module. Or in the commercial world, a company's marketing group may determine that a large market exists for payroll tracking software. In either case, when the need for some capability is uncovered, the end users normally do not have software experience or background, but do know what the desired results are. In these cases, the end product is manipulated via an Operator-Machine Interface (OMI).

The goal of program management is to convert user/customer needs into an unambiguous set of requirements for the development team [Ref 8].

A quality program manager will facilitate the user/customer needs into requirements that can be coded. This process happens in one of two ways. The first is the direct procedure. Users convey in any number of ways their needs to program management, which in turn develops the formal requirements to which the developers code. The second is the indirect procedure. The users convey their needs directly to the programmers who rapidly develop a prototype, which the users can see and validate. This process can be iterative. Program management adjudicates between user and developer during the indirect process and develops formal requirements. However, the formal requirements serve mainly as a record of what has been performed [Ref. 13].

Figures 1 and 2 illustrate program management's role in both approaches to requirement extraction [Ref. 13].

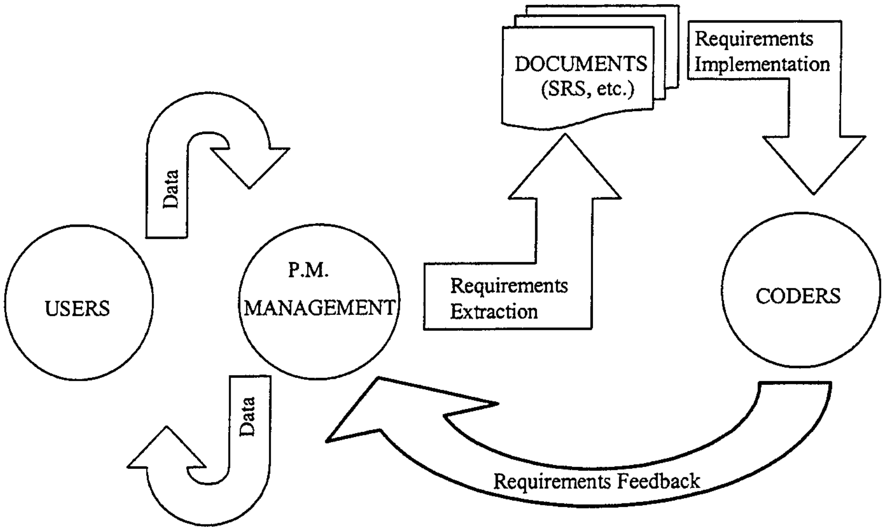


Figure 1. Determining requirements via direct program management involvement

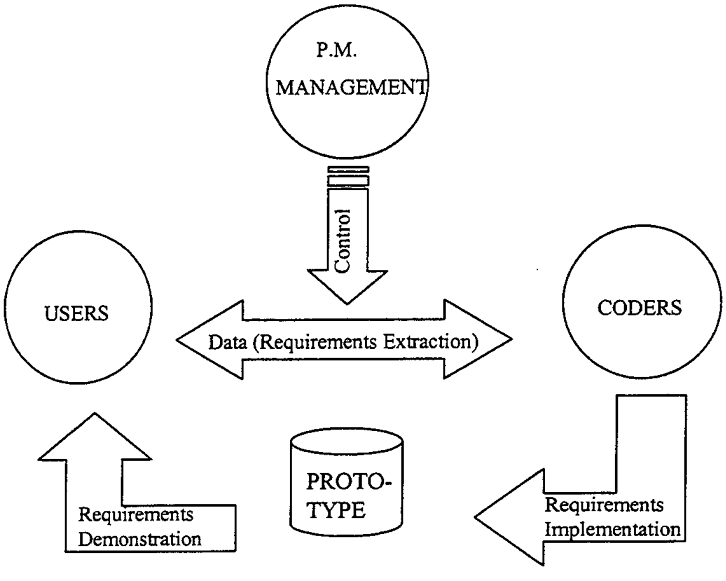


Figure 2. Determining requirements via indirect program management involvement

Many requirement definition techniques are available to aid the program manager. Use-case diagrams, Class Responsibility Collaborator (CRC) models, or other scenario type documentation is used to extract precise requirements. Because of past program failures due to poorly planned or derived requirements, consensus is that a program manager must enact some sort of formal process for the extraction and formulation of requirements. [Ref. 8]

CAPT Gerry Nifontoff (USN ret.) [Ref. 13] states

> ...because of today's tools, any software program involving OMI output, must involve direct dialog between the users and the developers.

The users express to developers what they need and the developers develop a quick prototype to feedback to the users. The process continues as program management facilitates and adjudicates the process.

Figure 3 shows possible actions a program manager can use to define requirements. It is Scott Ambler's [Ref. 8] "starburst" diagram for defining and validating initial requirements. This is an iterative process moving from center to one of the techniques and back again.

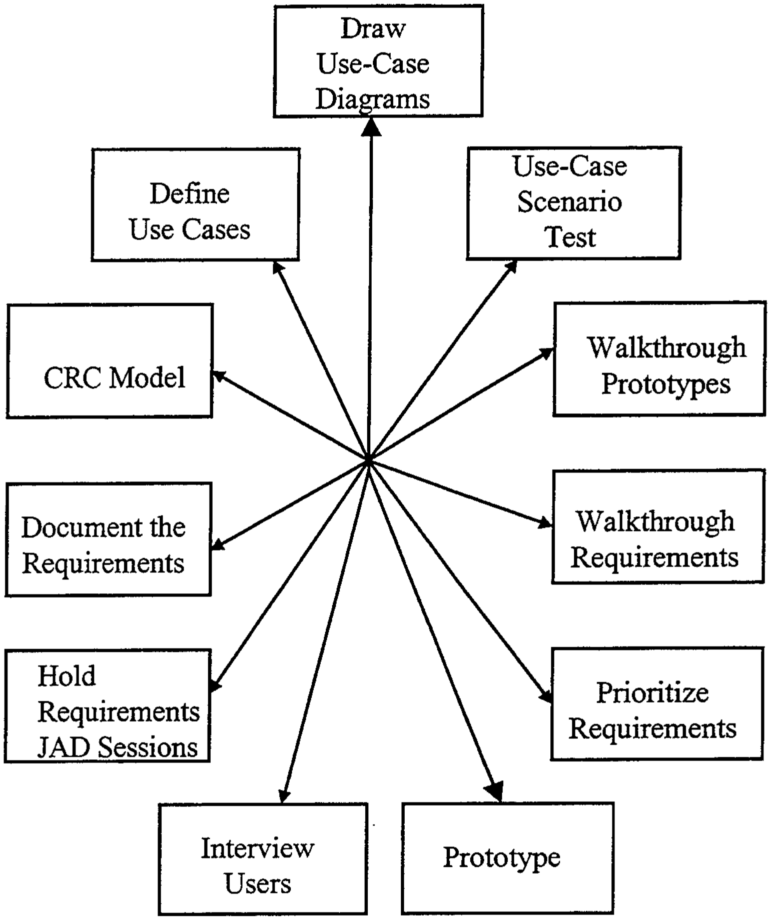


Figure 3. Starburst Diagram

This process, also called Rapid Application Development (RAD), is very popular today [Ref. 14]. Barry Boehm [Ref. 14] says that in general, RAD gives earlier product payback and more payback time before the pace of technology makes the product obsolete. For software product sales, RAD also helps debut a product earlier in a market window, which lets the product capture more market share, revenues, and profits. To gain the maximum benefit from RAD, the program manager must choose the RAD form that best suits the project.

Another closely related approach that is growing in popularity is throwaway software. This concept is simple. Upon startup, the developer may not know much, but while creating the software does learn what users really want and how to make clean code. By the time the project is finished, the developer has learned so much that it would be much better if everything was thrown away and started over. [Ref. 15]

The program manager's task is to analyze a project to find the hardest parts, then implement the throwaway software plan in these areas. [Ref. 15]

Synchronize and stabilize is an approach that companies such as Microsoft use to compete in the fast paced markets, such as Internet software. This model starts with a vision of what the product should do. The program manager derives a rough functional specification, which the team evolves until the end of the project. The schedule has multiple stabilization point, or milestones. Three is a common number. Each stabilization point represents progress after weeks of a development sub-cycle and usually represents an alpha or beta release. Requirements are finished when the development is finished and the product has been released. [Ref. 16]

The requirements list alone is not sufficient. It is the responsibility of the program manager to establish requirements prioritization [Ref. 17]. Time and money limitations apply, and decisions must be made on which requirements take precedence over the others. The program manager must ensure that a thorough assessment of all tradeoffs has been made. Outside factors play an important role in determining the options a program manager has in this area. On one side of the spectrum, a program that is limitless in funding and time can afford the program manager the maximum array of options. In reality, restrictions on both time and money to complete a development.

Identifying *all* the requirements upfront and then developing the product is idealistic in today's software environment. Requirements change for many reasons [Ref. 18]. It is the program manager's responsibility to establish some type of change management. Change management will help you direct and coordinate those changes so they can enhance - not hinder - your software [Ref. 18]. The change management procedures must be easy to understand and consistent. That is not to say that the development is subject to requirement changes at any time during the development. It is well documented that time and cost increase exponentially when requirements are changed late in the development process [Ref. 4]. The program manager must choose to "freeze" requirements at some point, but establish the framework for a follow-on version release or block upgrade. The lesson learned from the past ten years has been that software products are unlike most durable goods in that they are always changing. For instance, when buying or learning to use a new program or word processor, the user touted the view that the system would be long-lived. The user now desires and expects updates or new programs with added features and capabilities fielded in less than one year, with the system having a relatively short, useful life.

B. QUESTIONS

The questionnaire for requirements management evaluates the program manager on establishment of procedures. The goal is to tailor the software development process (and its management) to achieve optimal results, satisfying user/customer wants and needs with minimal time and money expended. These questions do not seek to determine the quality of judgements on any specific decisions made. The thrust of the questions is to establish the structure, if any, laid out by the program manager in the area of requirements.

Each survey is designed to pertain to a single program. The pair choice and yes-no questions address three encompassing areas of requirements management. The top three areas are not clear-cut and may overlap. They are extraction, change, and testability.

1. Extraction

Extraction covers the broad area of who is involved in the process, what the processes are, and when it is done. Customer dissatisfaction and cost overruns are often caused by poor requirements that are produced by people who do not understand the requirements process (or choose not to implement one) [Ref. 3].

2. Change

All programs have requirements change, with the sole exception being pure standalone, throwaway software. These questions ascertain how change is handled. Are there any procedures and what is the potential for creating stable changes for the system?

3. Testability

What is the program manager's view of testing requirements, and where is the emphasis placed on testing? Does the program manager consider testing up front or towards the end? Each requirement should be testable [Ref. 19].

Additionally, these questionnaires deal with formality. Formality determines how precisely requirements are explored, extracted, and recorded. Is the change process well defined? And has the test process been thoroughly defined? Whatever processes are used in the program, they must be well understood, recorded, and retrievable.

Figure 4 graphically illustrates the hierarchy of factors for requirements management.

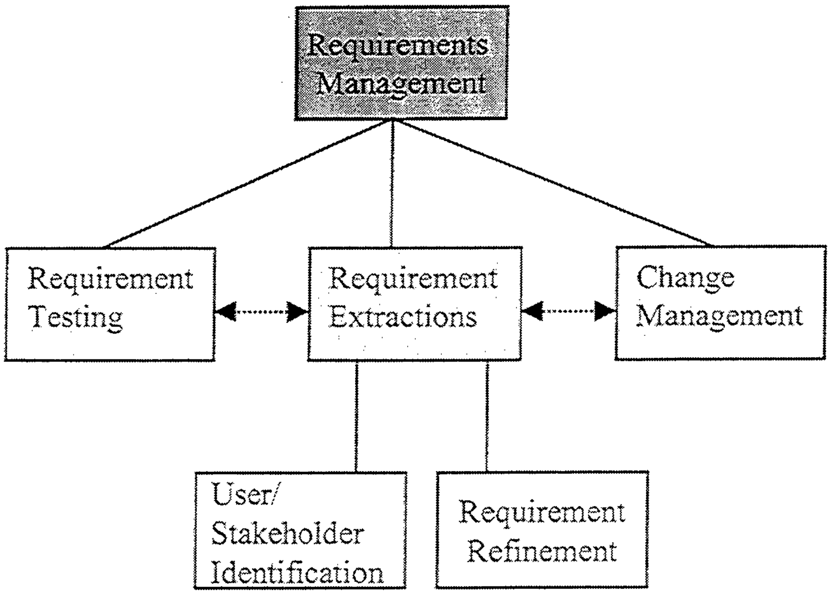


Figure 4. Requirements Management Hierarchy Factors

The three areas directly under requirements management indicate the next lower tier of factors to evaluate. The dotted lines between requirement testing, requirement extractions, and change management indicate the iterative relationship between the areas as a program progresses. Below requirement extractions are the activities of user/stakeholder identification and requirement refinement necessary for successful and thorough completion of requirement extractions.

III. ESTIMATION/PLANNING MANAGEMENT

A. COMPONENTS AND CRITERIA

Planning is the key to control. (Rick Weber [Ref. 20], Time Management Essentials)

When one thinks about management, one thinks about planning. Managers plan strategy, schedule, costs, etc. Software development programs and planning have been an oxymoron throughout the 1960s, 1970s, and early 1980s. Among software systems delivered, many were subject to cost overruns, late delivery, lack of reliability, inefficiency, and lack of user acceptance. [Ref. 21]

The basis of planning is estimation. Planning a software product development requires a frame of reference and an ability to measure against the reference. The program manager has three major measures to estimate the program by products, processes, and resources [Ref. 22]. Humphrey [Ref. 4] states,

> You measure to get data, and you want data to help you with the following:
>
> - Gain qualitative understanding
> - Evaluate a product, process, or organization

- Control a product or a process
- Make an estimate or a plan

Product measures generally refer to volume. Examples include lines of code (LOC), pages of documentation, number of screens, and number of files. The measure can be the whole product or various product elements, such as modules, components, or manuals. Measurement is accomplished by phase, such as the amount of code produced in the implementation phase or the LOC changed during unit testing. Measures of other product attributes might include system throughput, memory capacity, cyclomatic complexity, module coupling, and function points (FP). [Ref. 4]

Process measures quantify behavior, strategies, and execution of the process used to develop the product. One general category of process measures is event counts. Examples include the number of defects found in test, requirement changes, or milestones met. Another general category concerns time measures. The time required to complete a project is often called cycle time. In highly competitive markets, cycle time, or deployment, may be more important than reducing development costs [Ref. 4]. All the stakeholders and the organization must be considered and included in the analysis, planning, and implementation needed to release software.

Resource measures refer specifically to labor hours required for product development [Ref. 4]. Boehm [Ref. 1] further extends the measurement to include factors such as proper number and assignment of people to the work, the proper working conditions and reward structure for people, the proper resources (terminals, support software tools, etc.), and other quality management practices associated with requirements management and risk management. Pressman [Ref. 22] includes money as a resource measure. But money, unless it is a pre-set, fixed and known resource, cannot be properly included. Cost (money) typically becomes an estimated outcome from process, product, and resource measures.

Estimation utilizing all three measures for a program will lead to realistic planning of schedules and costs. Subsequent tracking of metrics throughout the program will aid program updates and provide a solid basis to which future programs can planned against [Ref. 1].

A simple example of using estimations to provide an initial program plan is the LOC a programmer can code per day. Estimate the product size and number of programmers, and duration estimates can be determined. Include the salaries of the workers over the duration, and cost is determined. With duration and cost estimated, an initial program schedule can be formulated.

Brooks [Ref 23] concedes that cost does indeed vary as the product of the number of men and the number of months, but emphasizes that progress does not. Reasons include the inability to adequately partition tasks because of sequential constraints and poorly drawn lines of responsibility due to management misjudgment. Poor correlation of consistent actual results also stems from the difficulty in estimating the productivity of programmers [Ref. 22]. It is estimated that differences in productivity among the best and worst programmers are commonly 10 to 1 and may be as high as 25 to 1 [Ref. 19].

Even when tasks can be partitioned, the burden of communication must be added to the amount of work to be done, particularly the effort required for intercommunication. If each part of the task must be separately coordinated with each other part, the effort increases as n * (n-1)/2, where n is the number of people needing to communicate. [Ref. 23]

Couple the productivity variances with other factors such as work environment, organizational structure, reward/recognition, training, and motivation, and the importance of management quality becomes very apparent.

For large, complex software programs, a Work Breakdown Structure (WBS) is recommended [Ref. 12]. A WBS defines all important tasks, milestones, and deliverables throughout the program [Ref. 22].

Once initial costs and schedules are derived from estimations, progress tracking and schedules and costs adjusting become key factors in the software program success [Ref. 12].

Establishing and tracking earned value is recommended to measure program progress [Ref. 12]. By assigning value to a developer's work package, the cumulative value of completed work packages can be compared to the estimated and actual cost to complete the work packages to give a more accurate measure of schedule and cost progress [Ref. 19]. Adjusting schedules and costs later in the program may appear to be an admission of failure of the initial planning effort. But Launi [Ref. 6] states that a universal truth applies to any project: change will occur constantly, dynamically, and usually, without warning. No matter how stable the initial estimates and plans seem to be, change occur as the program progresses for many reasons including discovery of unknowns associated with the product, process, or resources. It is a measure of software management quality as to how the changes are handled [Ref. 12].

The program manager must set up a structure to use product, process, and resource measures in a software program, and it is the program manager's responsibility to ensure that the measure being used will yield the most accurate and useful results possible for the software program.

B. QUESTIONS

The questions in this section ascertain that the program manager is performing both initial and follow up estimation and planning. The questionnaire checks that derived documentation is completed and used in the program. Moreover, it is used to determine if currently accepted methods and practices are being employed. Is the program manager managing the estimation and planning process sufficiently to give confidence to the product, process, and resource measures gathered? No attempts were made during interviews with program managers or discussions with focus groups to determine which measure or method is superior. The questions are designed to solicit the best structure and its practices.

Figure 5 graphically illustrates the main hierarchy factors in estimation/planning management.

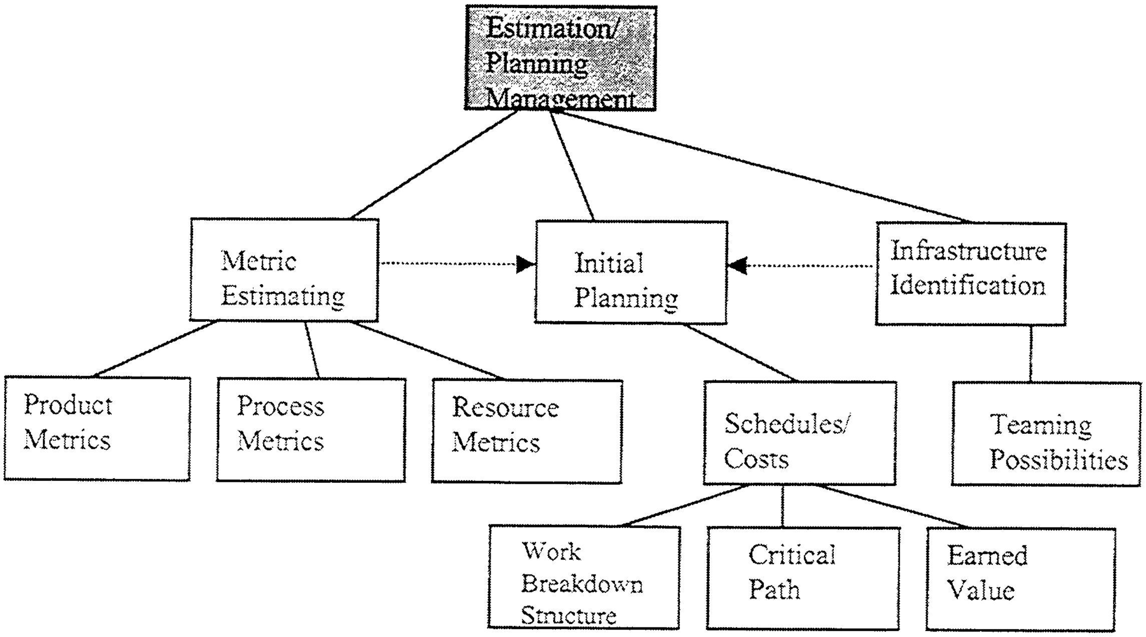


Figure 5. Estimation/Planning Management Hierarchy Factors

THIS PAGE INTENTIONALLY LEFT BLANK

IV. PEOPLE MANAGEMENT

A. COMPONENTS AND CRITERIA

The people management section of the thesis evaluates the software program manager in two ways: the skills that the software program manager exhibits and the type of organizational structure instituted by the program manager.

If a single person could perform all of the programming and software work on a product, there would be no need for people management. How management recruits, organizes, and treats human resources is instrumental to the success or failure of any endeavor involving many persons [Ref. 22]. Software development requires highly skilled professionals. Unlike producing widgets, software is a product of the mind. Although automated tools aid the developer, software is still largely based on individual interpretation and implementation.

1. Human Resources

The program manager must recruit, train, allocate tasks and teams, and reinforce positive behaviors for an overall working environment that increases a program's chance for success [Ref. 12]. Techniques that foster such an atmosphere includes showing appreciation, injecting humor whenever possible and empowering team members [Ref. 24]. In some organizations, particularly those like the Department of Defense, factors may limit the ability of the program manager to recruit, select, or otherwise change the software development team members. Restrictive organizations necessitate the program manager maximize existing human resources by concentrating on activities such as training and reinforcing positive behaviors to create a successful program environment [Ref. 13].

Training is often seen as a frill in many organizations, something to be reduced in order to meet profit goals in times of economic stringency. However, training can be a source of competitive advantages and is an integral component to an overall productive management practice [Ref. 25]. Software development programs with tight, hectic schedules are not an excuse for elimination or necessarily a good reason for postponement of training [Ref. 12]. The program manager must carefully plan training into the framework of the overall program schedule to ensure the organization of its long-term benefits without endangering short-term program needs [Ref. 25].

Luthans and Stajkovic [Ref. 26] state,

> The real challenge (of software program management) is to find ways to manage human resources as effectively as possible in order to attain world-class performance.

Reinforcing for performance is a tool the program manager can utilize to promote positive behaviors and eliminate negative behaviors. Organizational Behavior Modification (O. B. Mod) is a systematic approach based on reinforcement theory. Reinforcement theory's basic premise is that human behavior is a function of contingent consequences. Something that strengthens and leads to an increase in the frequency of a behavior is called a reinforcer, not a reward. Software program managers may not get desired behavior from individuals with pay and rewards alone. By reinforcing using O. B. Mod procedures, one always increases the strength and frequency of the desired functional, performance-related behaviors. Therefore, performance improvements will always result from reinforcing for performance. [Ref. 26]

O. B. Mod consists of five steps: identify, measure, analyze, intervene, and evaluate. The approach seeks to identify the critical observable performance-related behaviors, measure the baseline frequencies of the critical behaviors, analyze to determine the antecedents and contingent consequences in the performance-related context, intervene to increase the frequency of the positive behaviors and decelerate the dysfunctional behaviors, and finally, evaluate for performance improvement. [Ref. 26]

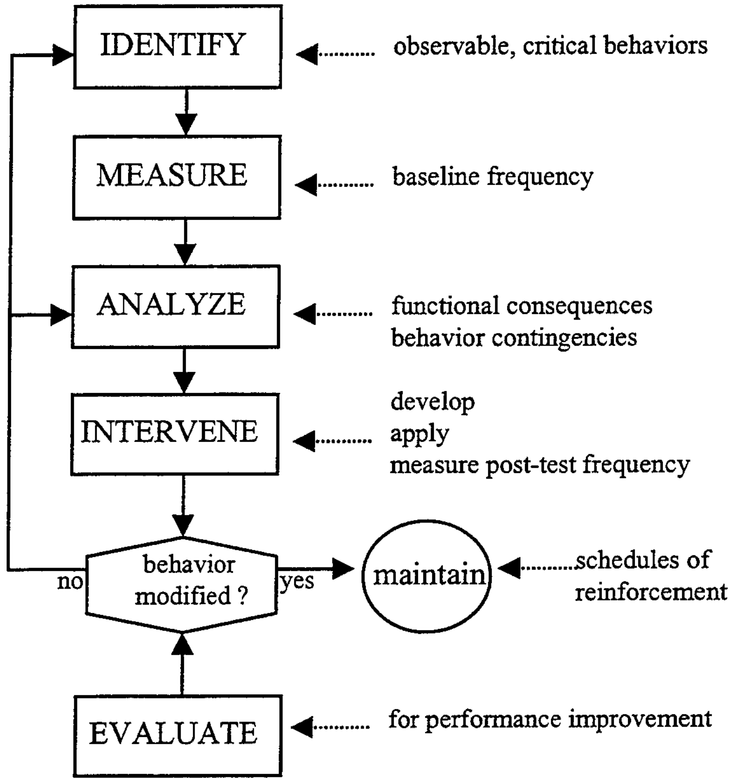


Figure 6. Adaptation of O. B. Mod application model

The use of the O. B. Mod application model is demonstrated in a service-sector example. Bank supervisors used contingently administered feedback and social recognition and attention reinforcers for teller customer service behaviors. This included using the customer's name, providing a balance, and making eye contact. These behaviors led to increases in measured customer satisfaction. In this same bank, the earlier use of monetary rewards had had no measurable effect on customer satisfaction. The money turned out to be a reward; not a contigently administered reinforcer that strengthened teller customer service satisfaction. [Ref. 26]

A corollary example specific for software development could focus on reduction of individual programming errors, a primary factor in determining software product quality [Ref. 27]. By identifying and measuring the critical behaviors that programmers demonstrated when writing good, error-free code, program management can then analyze the behavior contingencies and develop an intervention strategy. The program manager can then use one or more of the three types of reinforcers; money, feedback, and social; to promote the behavior leading to fewer errors in delivered code. The program manager can evaluate this improvement in performance against measures of costs and schedule. Reduced program costs and meeting schedule dates are direct results from reducing programming errors [Ref. 28]. Therefore, it is concluded that the use of reinforcers can help the software program manager effectively manage human resources to achieve desired behaviors and results from the software development team.

To date, improvement programs for software organizations have often emphasized process or technology, not people [Ref. 29]. The Software Engineering Institute's (SEI) People-Capability Maturity Model (P-CMM) was patterned directly after the SEI CMM structure. While the CMM focuses on software processes and practices, the P-CMM concentrates on a software organization's human resource management and development. The purpose of the P-CMM is to improve a software organization's ability to attract, develop, motivate, organize, and retain talent needed to steadily improve software development capability, via encouragement to meet high activity level standards. [Ref. 29]

As with the CMM, the level one for P-CMM is the initial level, the ad hoc activity level. Level 2 seeks to instill basic discipline into workforce activities. In level 3, management identifies primary competencies and aligns workforce activities with them. Level 4 has quantitatively managed organizational growth. Competency-based teams and practices are used.

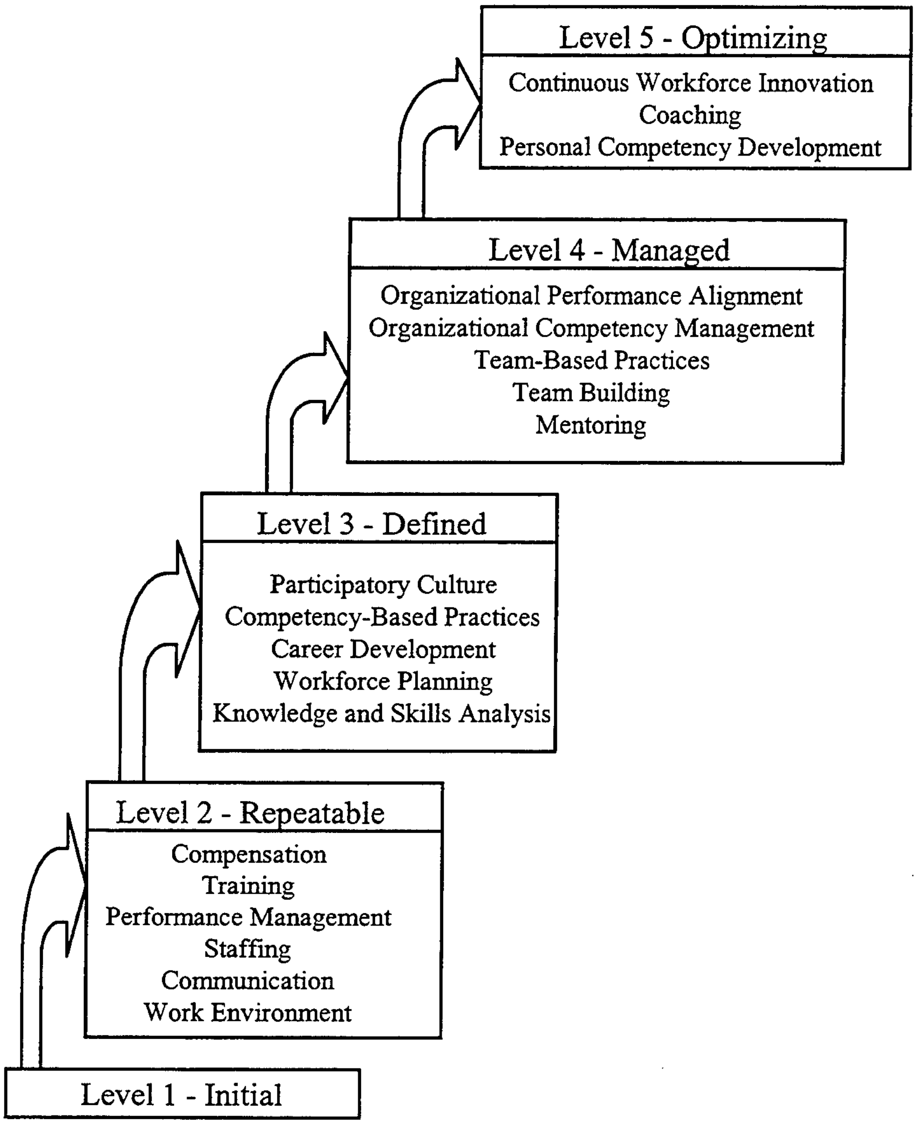


Figure 7. The Five Maturity Levels of the P-CMM [Ref. 29]

Team building skills are utilized and mentors are made available to both individuals and teams. The fifth level is the optimizing level where there is a continuous focus on improving individual competencies and innovative ways to improve workforce motivation and capability. Coaching, more formalized and greater depth assistance is provided to both individuals and teams. The organizational culture is created and evolves, as all members of the workforce are striving to improve the individual, team, and unit knowledge, skills, and motivation. [Ref. 29]

P-CMM is concerned with the issues that primarily come under the human resources section of people management. Over levels two through five, twenty key process areas are described. However, those twenty key process areas roughly cover four general areas: individual motivation, individual development, team forming, and team development. The result is an organizational culture. An organization's culture is manifest when its members share core values that guide their behavior [Ref. 30]. An organization that lacks repeatable management or development practices does not have a culture [Ref. 30].

2. Leadership

Software managers have the crucial role in establishing culture. Leadership from software managers comes before process or organization and the capability model makes no overriding differentiation [Ref. 31].

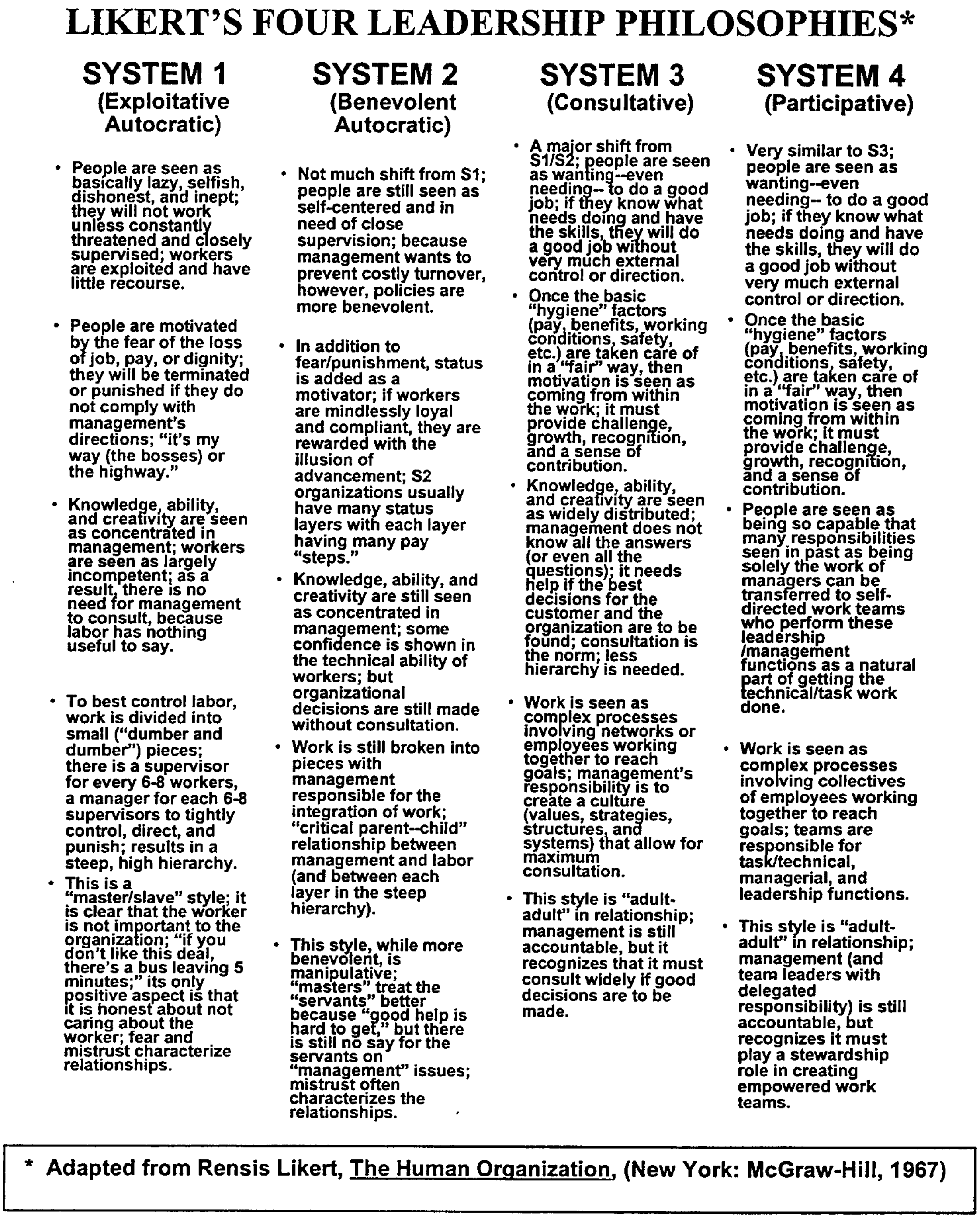
Therefore, software program managers are responsible for providing the leadership to enable good practices for managing human resources. While there are many different styles and personalities involved in management, each with its own strengths and weaknesses, a cross section of positive behaviors have been identified [Ref. 32]. These behaviors are based on the Myers-Briggs Type Indicators (MBTI). MBTI was developed from the psychological type theory work of Carl Jung [Ref. 33].

The four scales, each with two opposite poles, broadly covers all areas that a manager would be characterized. The four areas are: attention focus (Extrovert vs. Introvert); information gathering (Sensing vs. Intuition); decision-making (Thinking vs. Feeling); and orientation towards the outer world (Judging vs. Perceiving). Based on combinations, there are sixteen distinct patterns of behaviors defined. The MBTI survey is devised as a repeatable objective view on where the tendencies of a person lay. A series of questions is presented with a choice between two words or phases that best describe the preference. Based upon the totals, the preference is mapped onto the respective scale for each area. [Ref. 33]

There is no right or wrong judgement associated with the MBTI scale preferences. The preference identification is meant as an evaluation of where individual strengths and weaknesses lie. Street [Ref. 34] believes that leaders whose Type Indicator preferences tend toward any of the sixteen personality preferences in the MBTI can be successful. Each personality should work to expand the natural positive type traits and minimize the negative traits, or substitute more conducive, unnatural behaviors [Ref. 33].

Based on MBTI, Burgess and Street [Ref. 32] developed the Wave Model. The Wave Model defines five areas that a successful supervisor must excel in. These areas, in order, are personal skills, interpersonal skills, team skills, leadership skills, and organization skills. Successfully understanding and implementing each area successively builds upon the next, that is, organizational skills can be mastered only after the prior four areas are mastered. [Ref. 32]

Personal job satisfaction and subsequent productiveness relies more on the micro-work environment than the macro-work environment [Ref. 13].



LIKERT'S FOUR LEADERSHIP PHILOSOPHIES*

SYSTEM 1
(Exploitative Autocratic)

- People are seen as basically lazy, selfish, dishonest, and inept; they will not work unless constantly threatened and closely supervised; workers are exploited and have little recourse.
- People are motivated by the fear of the loss of job, pay, or dignity; they will be terminated or punished if they do not comply with management's directions; "it's my way (the bosses) or the highway."
- Knowledge, ability, and creativity are seen as concentrated in management; workers are seen as largely incompetent; as a result, there is no need for management to consult, because labor has nothing useful to say.
- To best control labor, work is divided into small ("dumber and dumber") pieces; there is a supervisor for every 6-8 workers, a manager for each 6-8 supervisors to tightly control, direct, and punish; results in a steep, high hierarchy.
- This is a "master/slave" style; it is clear that the worker is not important to the organization; "if you don't like this deal, there's a bus leaving 5 minutes;" its only positive aspect is that it is honest about not caring about the worker; fear and mistrust characterize relationships.

SYSTEM 2
(Benevolent Autocratic)

- Not much shift from S1; people are still seen as self-centered and in need of close supervision; because management wants to prevent costly turnover, however, policies are more benevolent.
- In addition to fear/punishment, status is added as a motivator; if workers are mindlessly loyal and compliant, they are rewarded with the illusion of advancement; S2 organizations usually have many status layers with each layer having many pay "steps."
- Knowledge, ability, and creativity are still seen as concentrated in management; some confidence is shown in the technical ability of workers; but organizational decisions are still made without consultation.
- Work is still broken into pieces with management responsible for the integration of work; "critical parent--child" relationship between management and labor (and between each layer in the steep hierarchy).
- This style, while more benevolent, is manipulative; "masters" treat the "servants" better because "good help is hard to get," but there is still no say for the servants on "management" issues; mistrust often characterizes the relationships.

SYSTEM 3
(Consultative)

- A major shift from S1/S2; people are seen as wanting--even needing-- to do a good job; if they know what needs doing and have the skills, they will do a good job without very much external control or direction.
- Once the basic "hygiene" factors (pay, benefits, working conditions, safety, etc.) are taken care of in a "fair" way, then motivation is seen as coming from within the work; it must provide challenge, growth, recognition, and a sense of contribution.
- Knowledge, ability, and creativity are seen as widely distributed; management does not know all the answers (or even all the questions); it needs help if the best decisions for the customer and the organization are to be found; consultation is the norm; less hierarchy is needed.
- Work is seen as complex processes involving networks or employees working together to reach goals; management's responsibility is to create a culture (values, strategies, structures, and systems) that allow for maximum consultation.
- This style is "adult-adult" in relationship; management is still accountable, but it recognizes that it must consult widely if good decisions are to be made.

SYSTEM 4
(Participative)

- Very similar to S3; people are seen as wanting--even needing-- to do a good job; if they know what needs doing and have the skills, they will do a good job without very much external control or direction.
- Once the basic "hygiene" factors (pay, benefits, working conditions, safety, etc.) are taken care of in a "fair" way, then motivation is seen as coming from within the work; it must provide challenge, growth, recognition, and a sense of contribution.
- People are seen as being so capable that many responsibilities seen in past as being solely the work of managers can be transferred to self-directed work teams who perform these leadership /management functions as a natural part of getting the technical/task work done.
- Work is seen as complex processes involving collectives of employees working together to reach goals; teams are responsible for task/technical, managerial, and leadership functions.
- This style is "adult-adult" in relationship; management (and team leaders with delegated responsibility) is still accountable, but recognizes it must play a stewardship role in creating empowered work teams.

* Adapted from Rensis Likert, The Human Organization, (New York: McGraw-Hill, 1967)

Figure 8. Likert's Four Leadership Philosophies

Likert's Leadership Philosophies [Ref. 35] define four distinct organizational working environments. Every organization can be categorized as one of these four systems (or some combination thereof). System 1 and 2 are closely related. The basic premise of system 1 and 2 is that the program manager makes all decisions, team members are not included in decision making. The team members may be valued for technical skills, but work is segmented into controlled pieces. The team member's relationship to management is more of a master-to-servant. Systems 3 and 4 are also closely related. The basic premise of systems 3 and 4 is that team members are, to varying extents, part of the decision making process. Team member responsibilities are not strictly segmented and relationship to management is more of an adult-to-adult type. [Ref 35]

Regardless of an overall organization system type, every program manager determines what system type the program will reflect (the micro-work environment). Focus group data indicates the overall organizational system status is a lesser factor on productivity of an individual when the program manager successfully implements system 3 or system 4 practices within the program team. [Ref. 36]

3. Communication

Communication is the highest single component in measuring the quality of software program management [Ref. 12]. Communication includes that of the program manager directly (vertical), the structure set up by the program manager for the development team (horizontal), and that with the stakeholders and users (external).

Loomis [Ref. 37] says,

> Unlike many other industries, the software business does not have large stores of tested, standardized parts to draw from in constructing new systems. Without standardization, communication of the details becomes even more essential.

Whether directly involving program management or others associated with the program, communications must be fostered and promoted by program management [Ref. 13].

Pickering [Ref. 35] describes and promotes the Network Talent Model as an alternative to the rigid Industrial Model. The Industrial Model was developed at a time when the workforce generally had lower education and performed tasks of much less sophistication. With well-defined and limited skill roles, the common notion is that technical persons do not need to perform management or leadership skills and that management persons do not require technical expertise. This is generally not the case today and certainly is not the case with software development programs.

In contrast, the Network Talent Model (NTM) depicts each individual in a group, team, department, or organization as possessing some necessary level of management, leadership, technical, and team skills. Work is more complex and individuals need to take greater roles than just their assigned tasks. Everyone takes ownership of the product or service and participates in the direction of their company or organization's future. [Ref. 38] Present day software programs are highly complex and necessitate communication at all levels [Ref. 12].

Hierarchical structure exists in both models, but the NTM will vary in the specific levels of leadership, management, and technical responsibilities. A top-level individual has different leadership, management, and technical requirements and responsibilities than a lower level individual. Individuals participating in a software program are more productive in a Network Talent Model than an Industrial Model [Ref. 38]. Figure 9 illustrates the roles of individuals in the respective models. [Ref. 35]

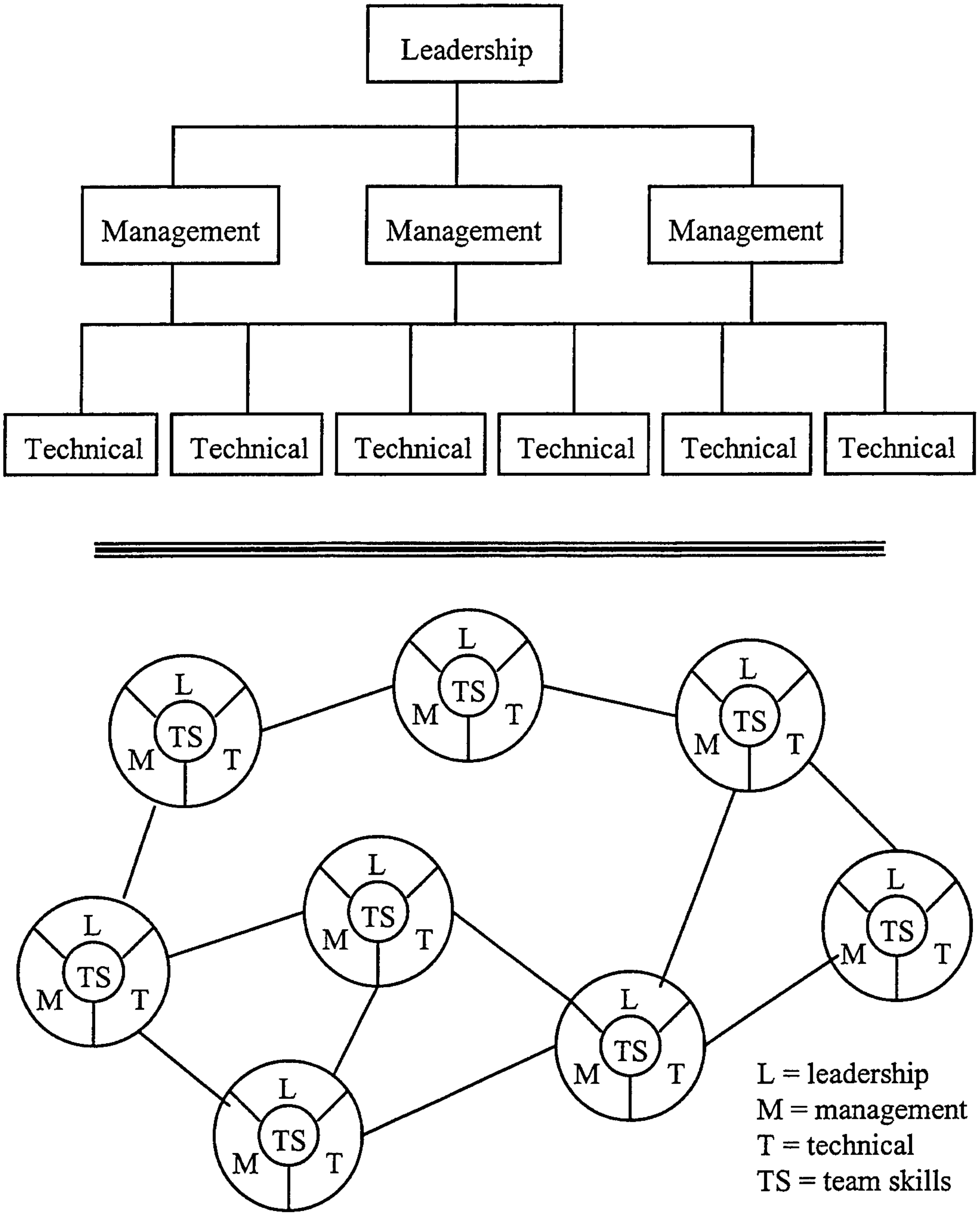


Figure 9. Role types in Industrial Model (top) and Network Talent Model (bottom)

Although the program manager may be hampered by the overall organizational structure regarding vertical communication upward, the program manager is responsible for ensuring effective internal horizontal communication among team members and internal vertical communication between team members and the program management. To the extent possible, the program manager should also foster the external communications among users, stakeholders, program management, and development team members [Ref. 12].

The challenge is to encourage open lines of communication, while residing within an organizational structure. Individuals vary in their ability to communicate; actions taken by the program manager will either improve or worsen the natural communication tendencies of individuals and teams.

B. QUESTIONS

Because the people management section encompasses many distinct areas that are highly weighted in importance, the questionnaire is divided into three sections; human resources, leadership, and communication. Questions are directed for consideration of human resource management. The leadership questions reflect the personal leadership skills exhibited and the leadership mentoring provided by the program manager. The communication questions seek to ascertain the communication protocols set up for the program organization and used individually by the program manager.

Overall, the questions do not attempt to type the program manager. Since the people management section is paramount to determining management quality, these questions seek to survey and query for the more conducive structure needed for a successful software program manager. Figure 10 illustrates the hierarchy of factors in people management.

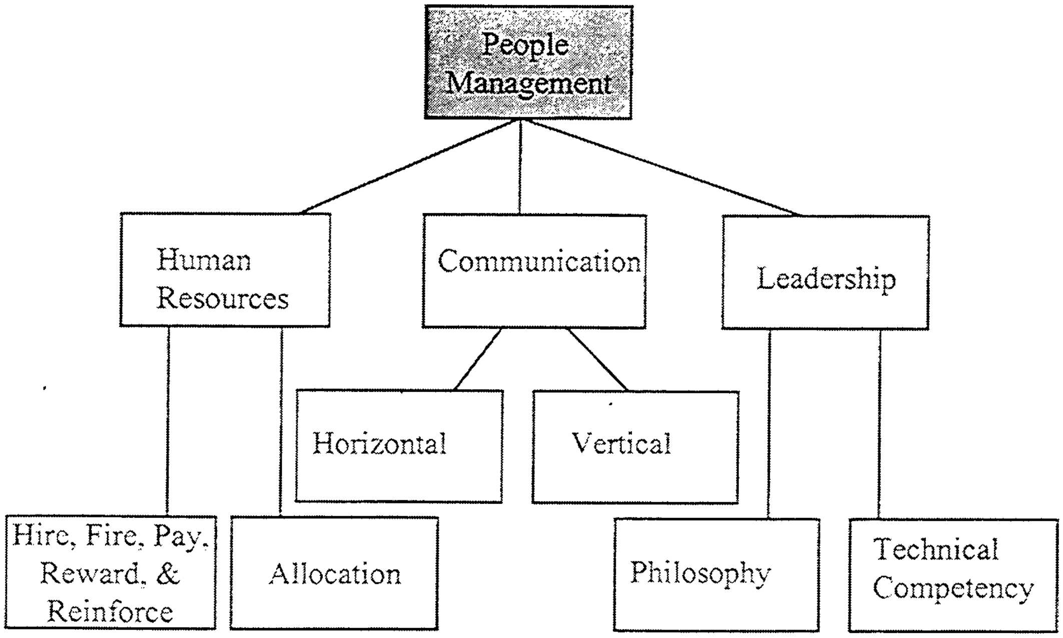


Figure 10. People Management Hierarchy Factors

THIS PAGE INTENTIONALLY LEFT BLANK

V. RISK MANAGEMENT

A. COMPONENTS AND CRITERIA

Wiegers [Ref. 7] defines risk as a problem that has not happened yet but could cause some loss or threaten the success of one's program if it did. These potential problems might have an adverse impact on the cost, schedule, or technical success of the program; the quality of products; or team morale. Because no program has ever run exactly as planned, every software program carries with it some degree of risk [Ref. 39]. Therefore, requirements management, estimation/planning management, and people management all contain risks.

Uncertainty is the unknown of what lies ahead. Attaching probabilities to future events changes uncertainty into risks. [Ref. 39]

Risk management is the process of identifying, addressing, and eliminating potential problems before they can do damage [Ref. 7]. It is included as a separate section and separate factor in this thesis because it is critical in measuring the management quality of a software program [Ref. 12, 13].

Figure 11 is the SEI risk management paradigm that defines a continuous set of activities that must be undertaken to identify, communicate, and resolve software risks [Ref. 40].

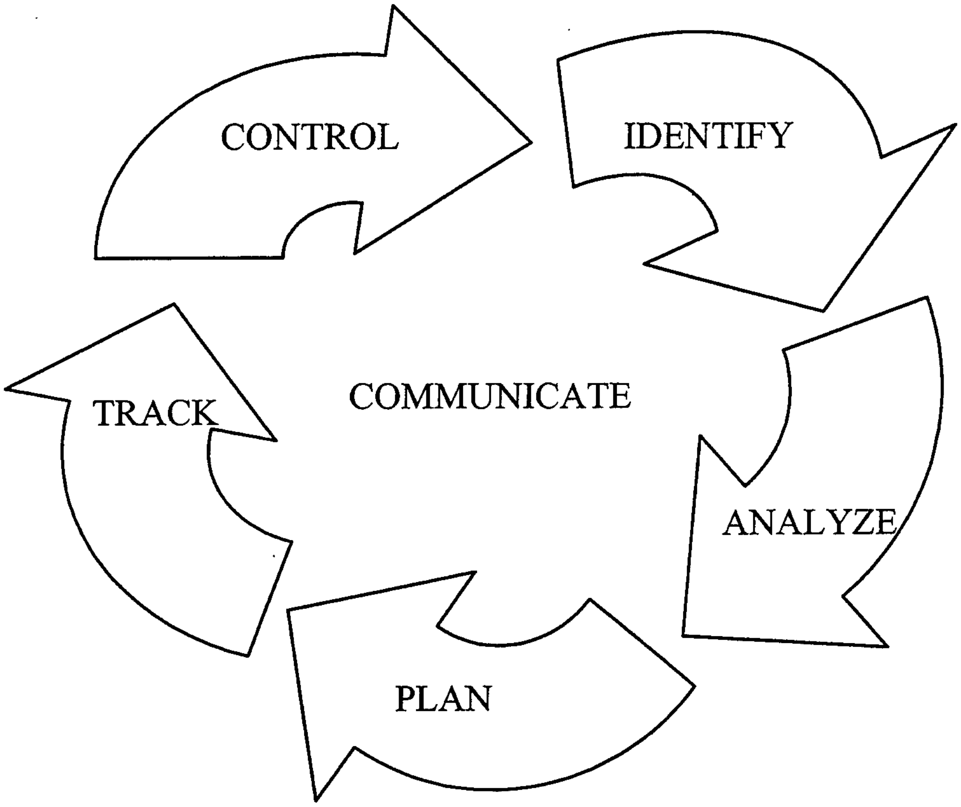


Figure 11. Risk Management Paradigm [Ref. 40]

1. Risk Assessment

Risk assessment is the action of examining a program and identifying areas of potential risk. Risk assessment encompasses the tasks of risk identification, risk analysis, and risk prioritization. [Ref. 7]

a) *Identification*

Identifying risks must be done individually. Keuffel [Ref. 39] classifies both macro and micro risks. The macro risks are used to measure the probability that specified tasks will be completed on specified dates, or that specified functionality will be contained within the product under construction. It compares the project's potential benefits with the overall costs and risks required to reach completion.

The micro view of risk management is the process of breaking a project into its component parts and identifying each variable. Since this is a painstaking process, Keuffel [Ref. 39] suggests injecting logic by considering the normal distribution range that each variable may occupy, not the possible range. This delineation would include risks of the lead programmer leaving to work for a competitor and exclude the risks of the lead programmer being struck by lightning.

Although each program is unique, the program manager can use history of similar size programs to identify risks. The use of the Software Engineering Institute's (SEI) checklist of possible risk factors or an organization's internal list is another good choice as the program manager considers checklist-based evaluations. [Ref. 7]

Although both practices are utilized, in risk management, the bottom-up approach is viewed more favorably than any top-down evaluation [Ref. 39]. Following this line of reasoning, program managers that hold team sessions and get people involved in developing the product to participate in risk management tend to have a better perspective on the risks associated with the program.

b) Analysis

Risk analysis involves examining how your program outcome might change with modification of risk input variables [Ref. 7].

c) Prioritization

Risk prioritization helps to focus the program on its most severe risks by assessing the risk exposure. Exposure is the product of two factors: the probability of incurring a loss due to the risk, and the potential magnitude of that loss. [Ref. 7]

2. Risk Control

Risk control, although listed separately in the SEI Risk Management Paradigm, encompasses risk planning, risk tracking, and risk resolution. Risk control is the process of managing risks to achieve desired outcomes [Ref. 7].

a) Planning

Risk planning involves developing actions to mitigate individual risks, prioritizing actions, and integrating them into an executable risk management plan [Ref. 40].

b) Tracking

Risk tracking involves monitoring the status of risks and their mitigation actions along with the use of metrics and triggering events [Ref. 40].

c) Resolution (Control in SEI model)

Risk resolution is the execution of the plans for dealing with each risk [Ref. 7].

3. Risk Communication

Communication refers to the exchanging of risk management information among the functions and at all levels of the organization. This activity is represented in the center of the model to emphasize its pervasiveness and criticality for implementing the other activities in the paradigm. [Ref. 40]

4. Risk Avoidance

Risk avoidance is one way the program manager can deal with a risk: do not do the risky things! You may avoid risks by not undertaking certain parts of the program, or by relying on proven rather than cutting-edge technologies when possible. [Ref. 7]

5. Regret Matrix

The Regret Matrix is part of the decision theory that further quantifies risk management by attaching probabilities to future events. This changes uncertainty into risk, which allows a calculation of net present benefit. Regret analysis performed on a risk evaluates potential actions the manager may take and its effect on the project. Impact effect scales are used; low, medium, high, in addition to some absolutes like no effect and program cancellation, to arrive at the best mitigation action to follow. In general, using actual measurements, like a function point count of 100, to trigger a program risk, yields to mathematical modeling and is perceived as more favorable than ordinal rankings of low, medium, or high. [Ref. 39]

The cost of resolving risk is relatively low early on, but increases as the program progresses [Ref.12]. The failure of the program manager to acknowledge and implement some level of risk management is an egregious error and objectively decreases management quality [Ref. 12]. Thus, quality management must include performance of some type of formal risk management. How well a risk management plan has been implemented can be determined in retrospect. The risk management factor of the quality management metric can only measure the risk management structure set up. The factor takes into account any structure that promotes success in the software development environment by considering individual risks, assessing individual impact, determining a probability of occurrence, and planning a mitigation strategy. Program management's judgements within the established structures will vary, and can ultimately determine the success or failure of a risk management effort. However, the establishment of structure dedicated to these practices can be objectively measured and yield a strong indication of the quality of program management.

An example of the importance of risk management: the SURTASS program had at least two dramatic external changes that changed the mission of the development program. First, in the mid-1980s, the Toshiba Corporation of Japan, sold the Soviet Union advanced milling equipment that enabled the

Soviet Union to produce quieter submarines. The program requirements changed significantly as the focus shifted from passive to active functionality. Secondly, in the late 1980s, the collapse of the Soviet Union dramatically altered the mission need of the program and impacted the planned production. The goal of risk management is to anticipate these possible risks and have mitigation plans in place for necessary alterations to the development program [Ref. 13].

B. QUESTIONS

The questionnaire in this section will ascertain the structures used by program management for identification, monitoring, and managing risk. The questions determine whether the program manager has set in place strategies and personnel to thoroughly implement risk assessment, explore, and prioritize all reasonable risks. Does the program manager have an active risk management program and established procedures to monitor the risks and update the plan? The goal is to ensure that the program manager has, for each identified risk, an integrated mitigation strategy.

Dependence on strict methodology (notes, lists, and spreadsheets) alone for assessing risk is viewed poorly, while simple spreadsheet tracking along with thorough risk analysis is viewed more favorably. The overarching idea with identifying risk is that while a structured approach is helpful and necessary, the very human input, such as thinking between the lines, uncovering the unexpected, and an ad hoc approach, is also necessary to get a complete and thorough risk assessment. [Ref. 12]

Besides initial risk assessment planning and establishment, another important factor is how program managers implement it throughout the program's development. Is the Risk Management Plan put away and counted merely as a data call satisfied, never to be used again? Or is the Risk Management Plan implemented, revisited, and updated?

Figure 12 graphically illustrates the risk management hierarchy of the activities evaluated in the risk management component of the quality management metric.

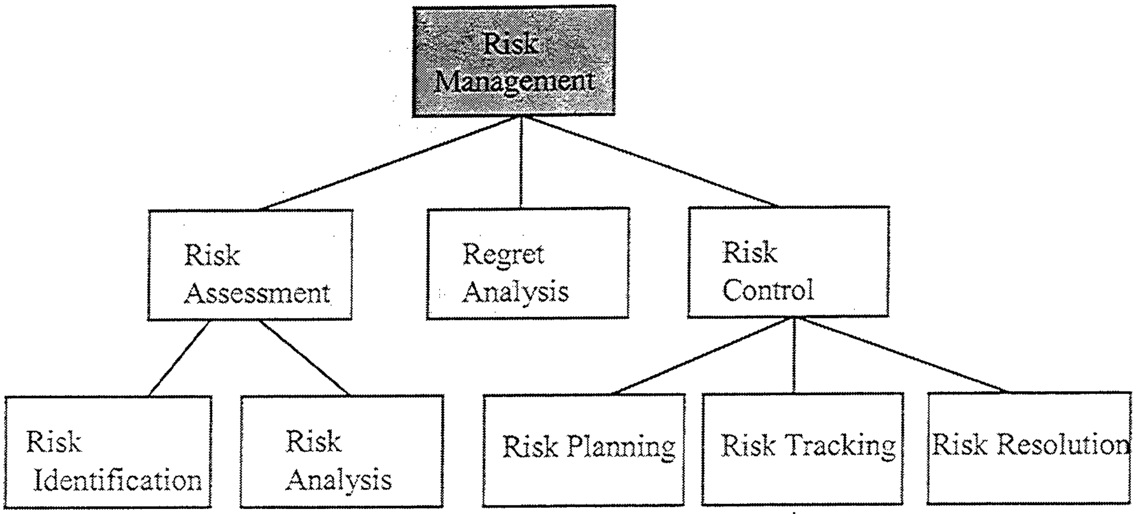


Figure 12. Risk Management Hierarchy Factors

Any risk management plan is useless unless it is updated along with the software program's changing environment. The constantly changing environment from organizational strategy, competitive pressures, changing political landscape, technical challenges, and personnel changes, may dramatically alter a program. [Ref. 12]

It is difficult to measure individual judgements about risk management. What can be measured is whether the program manager has performed risk management elements.

VI. CONSTRAINT FACTORS

Constraints are factors limiting the options that the program manager has in executing the program. The program manager's quality score should not be impacted by actions that are not within the program manager's control. For a software development program, the two main sources of constraints are those imposed by the company or organization itself and those from the stakeholders of the program [Ref. 12]. Money and schedule are typically how constraints are imposed [Ref. 41]. In other cases constraints may be a mandated architecture, development model, operating system, or suite of development tools (e.g., compilers, CASE tools, configuration control, and management tools). All software development programs contain constraints that the program manager must content with.

A. REQUIREMENTS MANAGEMENT CONSTRAINTS

Constraints in requirements management include: using requirements extracted by other groups, no control of requirement implementation, no prioritization flexibility (all requirements are priority one), and little to no interaction with the users. One of the most frequent constraints facing program managers is not being able to limit requirement changes during the program execution [Ref. 41]. Even with a rigorous change management structure, stakeholders can and do dictate circumvention of the process to facilitate their desires or changing needs [Ref. 12].

B. ESTIMATION/PLANNING MANAGEMENT CONSTRAINTS

Money and time are the main constraint factors in the estimation and planning activities of a software development project and therefore must be treated as resources that are to be managed. Program managers are often forced to buy in to programs that are either inadequately funded and/or have unrealistic schedules [Ref. 12]. Frequently money, time, or both strictly define the capabilities of the software product being developed.

Other constraints include stakeholder-mandated use of specific metrics for estimating. Applying different metrics can yield different estimation results, therefore the mandated choice of a particular metric on which to base estimations can influence planning, and thus execution of a program. Boehm [Ref. 1] further adds,

> Having a good software cost model available does not guarantee good software cost estimates. As with other computer-based models, a software cost-estimation model is a "garbage in-garbage out" device: if you put poor sizing and attribute-rating data in on one side, you will receive poor cost estimates out the other side.

The implication is that certain types of software developments are better suited to certain metric estimation models than other programs are. The program manager must be afforded the opportunity to evaluate alternative techniques and compare their relative strengths and weaknesses. [Ref. 1]

C. RISK MANAGEMENT CONSTRAINTS

Risk management constraints primarily involve funding. Nifontoff [Ref. 13] states that risk management can be done cheaply or expensively. The cheap method is to rely on the existing senior program managers and engineers to perform risk management. The expensive method is to bring in outside consultants to perform risk assessment and mitigation.

Stakeholder views on the importance of and willingness to adopt and act on risk management recommendations influence the amount of funding allocated to the effort. Even if stakeholders refuse to fund risk management efforts as a separate line item, Nifontoff [Ref. 13] says the program manager must perform risk management,

> ...whether computerized or with wall charts or sitting around a table, it still must be done.

Consequences of not performing risk management can be devastating to a software program. Programs have failed even though all the other areas were sufficiently addressed because of failure to consider risks [Ref 41].

D. PEOPLE MANAGEMENT CONSTRAINTS

There are many possible constraint factors in people management. Most of these involve constraints imposed by the company or organization [Ref. 36]. The program manager may be unable to obtain qualified personnel or to release team members who do not fulfill program needs. The limitations on salary compensation, rewards, and bonuses can be more restrictive in Government organizations than commercial companies [Ref. 13]. Executing a software development program within an activity or company with an organizational structure classified as a system one or system two in the Likert model is a constraint [Ref. 36]. Pickering [Ref. 35] believes that the program manager must structure the software development team as a system three or system four to be successful. In this scenario, the constraint imposed by the overall organization must be overcome. Pickering [Ref. 35] adds that often, whole organizations change this way -- from the bottom up.

The lack of training provided by the organization is another constraint in people management [Ref. 36]. In most organizations, funding for training is separate from the specific program funding. Therefore the program manager may not have an ability to provide needed training for individual team members within the organization.

E. QUESTIONS

Questions exist in each of the four sections that help ascertain where program management may be constrained. In the yes-no-n/a questionnaire, the not applicable (N/A) selection is used for questions that do not apply to the program or for areas in which the program manager does not have control. The questions are designed so the quality management scoring will not be affected where constraints are present.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. METRIC METHODOLOGY

A. STRATEGY

The approach used to develop the metric included researching the successful current and recommended practices chronicled in textbooks and periodical publications, and obtained via both interviews with senior program managers and conducting focus-group meetings. The metric measures the quality of management on a specific software program. The overall goal is to develop an objective, standardized metric to which program management can be compared and ranked, thus providing a baseline for quantifying improvement. This metric compares the same management on two different software programs or at two different time intervals of the same program. Metric development is difficult because the quality of management can be very subjective. Words like feel, think, believes, etc. which prompt subjective responses are avoided as much as possible. Subjective answers are not useful in obtaining quantifiable, objective results. Answers are constrained to enable scoring to a scale. The technique used is a two-part questionnaire for each of the four sections.

B. QUESTIONNAIRE FORMAT AND SCORING

Questions and concepts used in the questionnaires were gathered and compiled from periodical articles, textbooks, interviews and focus groups. The concepts included are relevant to judging the quality of management. Participants taking the survey were asked to consider one software program at one particular instance of time.

Part one of the questionnaire contains pair choice questions. The person filling out the questionnaire must choose one of the two statements that best describe the program. The choice does not have to match exactly; it should just be the closest fit. The model for this type of questionnaire is the Myers-Briggs Type Indicator (MBTI) questionnaire [Ref. 33]. The format used in the MBTI questionnaire requires participants to choose between two statements. Each pair statement represents two differing ideas in an effort to ascertain a tendency of the individual. Often the pair choices are repeated with different wording to confirm earlier choices and measure the strength of the tendency. The survey format, with the proper mix of questions and variation repetitions is intended to be used to reach consensus on issues and measure the strength of tendencies. Each section has a maximum score of 70 points. The risk management, estimation/planning management, and people management sections have 70 questions each. The 70 questions in the people management section are apportioned according to the importance rankings of four of its lower-level factors. Some questions apply to more than one factor. The requirements management section has 50 questions and includes an alternate block of sixteen questions depending on the development strategy used.

Part two of each questionnaire is the yes-no-n/a questions. Instead of asking open-ended questions that participants could answer in a variety of ways in essay form, the yes-no-n/a format standardizes the responses for easier comparison. The yes-no-n/a format is user-friendly for conducting surveys, requiring minimum writing by the participant. Each yes, no, or n/a choice has a point value based on the relative importance of the question. Each section has a maximum value of 62 points. The estimation/planning management, people management, and risk management sections have 50 questions each. The requirement section has 47 including an alternative block of six questions depending on the development strategy used. The complete survey, including both parts for all four sections, contains 457 questions.

Administration of the questionnaire to each participant was conducted such that the subject was not given any information about the point value of each response; this was done to avoid any pre-bias tendency of one response over another. Manually scoring the questionnaire focuses attention on the entire process and de-emphasizes focusing only on the final Quality Management Metric (QMM) score.

The point totals from each of the two questionnaire parts per section are entered on the QMM Summary Score Sheet. Point totals for part one and part two are then added together to determine the total points for each section. The total points of each section are multiplied by its relative Importance Coefficient (IC) to yield a weighted score. After weighted scores are determined for each of the four sections, they are summed together to yield the Quality Management Metric (QMM) score.

The IC was determined from the relative rankings of importance of each of the sections. Experienced software professionals provided the data to determine the IC through the focus groups and one-on-one interviews only after thorough explanation and understanding of each category. A total value of forty points was allowed for allocation over the four sections.

Figure 13 is the summary of the raw data used to determine each section IC. The QMM equation is as follows:

$$QMM = 0.92\ RqM + 0.67\ EPM + 0.55\ RkM + 1.86\ PM$$

The QMM is the sum of four components:

RqM is the requirements management metric

EPM is the estimation/planning metric

RkM is the risk management metric

PM is the people management metric

Because of the overwhelming importance placed in people management, PM is further broken into four components that were individually ranked. The PM is the sum of its four components.

The four components are L, the leadership measure, C, the communication measure, HR, the human resource measure, and TC, the technical competency measure. Data for determining the IC in each of the four components of people management was gathered with the same methods used to determine the IC for the four management sections. However, the total points spread across the people management components could not exceed the total points allocated for people management.

The equation for People Management is PM = 0.65 L + 0.55 C + 0.41 HR + 0.25 TC.

RATED	Individual responses A through Q (40 pt must of 4 main categories)																	Importance
CATEGORIES	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	Avg	Coefficient
Requirements Mgmt	5	4	18	12	12	10	8	5	9	6	3	6	15	5	17	10	12	**9.2**	**0.92**
Est./Planning Mgmt	4	7	0	12	7	10	10	3	7	6	3	4	10	10	12	5	4	**6.7**	**0.67**
Risk Management	6	7	2	6	7	5	7	2	4	6	3	5	5	5	3	10	10	**5.5**	**0.55**
People Management	25	22	20	10	14	15	15	30	21	22	31	25	10	20	8	15	14	**18.6**	**1.86**
Human Resources	6	7	0	3	4	4	4	0	2	6	5	8	2	5	2	8	4	**4.1**	**0.41**
Leadership	7	4	10	3	4	4	4	20	9	10	10	8	3	5	2	2	5	**6.5**	**0.65**
Communication	7	4	10	3	4	4	4	10	9	6	10	8	3	5	2	2	3	**5.5**	**0.55**
Technical Competency	5	7	0	1	2	3	3	0	0	0	6	1	2	5	2	3	2	**2.5**	**0.25**

Figure 13. Importance Coefficient Development

The maximum QMM score possible for the entire survey is 528.00 points and the minimum possible score is -130.86 as part two questionnaires contain negative point response values.

VIII. INFORMAL VERIFICATION AND VALIDATION

A. MOTIVATION

The structure and methodology for evaluating the quality of software management laid out in the previous chapters is informally verified and validated in this section. The informal verification and validation is necessary to ensure that the metric measures the quality of software program management and that it does so as accurately as possible.

B. STRATEGY

The approach to verification and validation is informal. Three software programs were evaluated for a QMM score. The program manager and one program development team member evaluated program A. Program B was evaluated by the program manager and two program development team members, and program C was evaluated by the program manager and one program team member.

In order to provide a frame of reference in which to correlate initial survey results, two other measures were developed and used. These two measures are the QMM percentage score and the overall program success score.

The QMM percentage score is a derived measure of the QMM score. To obtain a QMM percentage score, the following steps are required. First, the survey minimum possible score is normalized to zero. Since the survey minimum QMM score possible is -130.86, 130.86 is added to the survey minimum possible score in order to have it equal zero. Correspondingly, 130.86 must be added to both the survey maximum QMM score possible and to the actual QMM score obtained in the survey. Since the QMM survey maximum possible score is 528.00, the resulting normalized survey maximum possible score is 658.86.

To obtain a QMM percentage score, the normalized QMM score obtained from the survey is divided by the normalized survey maximum possible QMM score and then multiplied by 100. Thus, the equations are as follows:

$$QMM_{MIN} + 130.86 = 0.00 = QMM_{MIN\ NORMALIZED}$$

$$QMM_{MAX} + 130.86 = 658.86 = QMM_{MAX\ NORMALIZED}$$

$$QMM_{SCORE} + 130.86 = QMM_{SCORE\ NORMALIZED}$$

$$(QMM_{SCORE\ NORMALIZED}/QMM_{MAX\ NORMALIZED})*100=QMM_{PERCENTAGE\ SCORE}$$

The overall success score is a subjective number assigned by the survey participant rating the overall success of the program. The success of a program is measured in terms of how well the final product performs and meets the expectation and satisfaction of users and stakeholders.

The survey participant's QMM score is compared to his or her individual overall success score and to the mean overall success score of the program. The mean overall success score of a program is derived from each survey participant's individual overall success score and at least two other individuals (mostly users, or those somehow associated with the program or delivered product) able to judge the overall success of the program.

The overall success score is measured on a scale of zero to ten. Zero is defined as abject program failure with no worthwhile product. Ten is defined as absolutely perfect software product associated with flawless program execution. The cause for success or failure of the program is not important. It may or may not be associated with any actions involving program management.

The QMM percentage score is compared to the individual and mean overall success score of the program.

The goal was to determine any correlation between the participants' QMM score, their individual success ranking of the overall program, and the mean success ranking of the overall program. For example, an overall success score seven corresponding to a QMM percentage score of 70 percent plus or minus 5 percent indicates strong correlation. An overall success score of seven corresponding to a QMM percentage score greater than plus or minus five percentage points of 70 percent, but less than plus or minus 15 percentage points of 70 percent is considered fair correlation. If a program has an overall success score of 8 corresponding to a QMM percentage score of 40 percent, this would be considered weak correlation. In this last case, the QMM metric is still valid as programs with high quality of software program management could conceivably fail for a variety of reasons, including poor technology or funding shortfalls. The reverse condition may also be true for explaining successful programs with low quality of software management. However, it is typically expected that successful software programs follow superior software program management.

C. RESULTS

After the survey was scored, a QMM was determined for the program. The QMM score is measured as a percentage of the maximum QMM score possible. That percentage was compared to the subjective assigned score of the relative success of the project to obtain a comparison basis. Table 1 summarizes the resultant scores of the three programs. The subscript PM indicates the program manager's survey results and the subscript numeral indicates a participant's survey results other than the program manager. The mean success score of a program includes the individual success ranking scores by the individuals participating in the survey plus others associated with the program in some way where they can judge the success of the program.

Program	Program A		Program B			Program C	
Participant	A_{PM}	A_1	B_{PM}	B_1	B_2	C_{PM}	C_1
QMM Score	338	322	386	106	47	198	189
QMM percent	71.2%	68.8	78.5%	35.9%	27.0%	49.9%	48.6%
Success score	7	7	9	4	3	4	4
Mean success score (0-10)	7		4			4	

Table 1. Results Summary Comparison

The survey results for program A and program C exhibit correlation between the QMM percentage ranking and the overall success ranking of the program, both with individual success ranking scores and the mean ranking score. The QMM summary sheets for each survey completed are enclosed as Appendix A. An examination of the summary sheets for program A reveals a weak risk management section. This conclusion appears correct, as risk management for this program was not emphasized. However, program A was highly structured and planned, involved key users well enough to warrant successful requirement extraction and enjoyed good technical success with their deliverables. The higher scores in the other three sections reflect this success.

Program C was a smaller program that was relatively unstructured, with essentially no risk management, little planning and poor requirement extraction. However, the program has delivered a usable product, primarily on the heels of strong practices in the people-management portion and a technology that was relatively straightforward and understood.

Program B exhibits a significant divergence from the scores of the program manager and the other team members. In this case the program manager's scores on both the QMM and individual success ranking were significantly greater than the mean success ranking and the QMM scores and individual rankings of the other two participants from program B. This program appears to have a dichotomy in perception. Further interviews with others in the program and users of the product indicate that there are some significant issues needing resolution.

Having a good method or model does not guarantee good results [Ref. 1]. Inaccurate or incomplete information will significantly affect the validity of survey scores. Additionally, the self-enhancement bias is a perverse social psychological phenomenon. Researchers have found that one of the most widely documented effects in social psychology is the preference of most people to see themselves in a self-enhancing fashion [Ref. 25]. On the job, approximately 90 percent of managers and workers rate their performance superior to that of their peers [Ref. 42]. Surprisingly, it is not only the answers to the more subjective survey questions that vary among participants in the same program, it is also some of the answers to the purely objective questions on the survey. These results not only make the case for the requirement of a survey administrator; it also points to a need for conducting interviews in addition to administering the survey to better judge the results.

Survey results that vary significantly between program management and team members can be very useful in uncovering significant differences in perceptions about what is thought to be occurring and required in a program and what is actually occurring and required in the program. Bringing the participants together after the survey has been completed and scored to discuss the significant differences in their answers could be the single biggest benefit of the QMM process.

The participants provided additional feedback and recommendations for improvements to the concepts surveyed in each of the sections and also for improvements in individual questions asked. Copies of the QMM summary sheets for all seven of the survey participants are included in Appendix A. Copies of the completed survey from each of the three program managers are included in Appendix B. The resultant survey questionnaire template with ranking of each response is included as the Appendix C.

IX. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

This thesis provides an initial structure and basis for evaluating software management performance on specific software programs. The goal of creating an objective, repeatable metric for determining the quality of software management was obtained. The quality of management on software programs varies considerably and is a significant element in the overall success of a program [Ref. 1]. The policies and decisions that the program managers make influence the success of a software program.

1. Top-Level Evaluation Sections

Individual software program managers vary in their style of running a program. Using the MBTI as a model, the thesis identified important characteristics of successful managers and rated them accordingly via the two-part questionnaire. The four top-level evaluation sections, requirements management, estimation/planning management, people management, and risk management encompass all activities and techniques used to execute a software program. Overwhelmingly, the people-management section was rated highest in importance in judging the software program manager's performance. Four lower-level factors, leadership, communication, human resources, and technical competency of the program manager were subsequently individually categorized and rated within the people management section alone. Focus groups and survey participant's results concurred that the people management factor dominates a software program manager's performance.

2. Survey

The survey format, length, and individual questions achieved the goal of covering the important processes and concepts involved in the quality of the software manager in an acceptable amount of time dedicated by the participants. The average survey completion time was approximately 45 minutes. The longest timed participant took almost 60 minutes and the shortest timed participant took approximately 35 minutes.

3. Metric Scoring

The comparison of the QMM percentage score to each individual overall program success score yielded strong correlation in each instance. The comparison of the QMM percentage score to the mean overall program success score yielded strong correlation in all but one instance.

Six of the seven survey participants recorded QMM percentage scores within 13 points of the mean success score percentage for their respective programs. This indicates strong correlation of the metric with program performance.

The one exception was the highest QMM score recorded at 386.15 and with a corresponding QMM percentage score of 78.5% on a program with a mean success rating of 4 exhibited a significant variance. However, that participant gave that same program an individual program success score of 9, which was also a significant variance from the mean success score of 4. This divergence indicates a significant difference in perception of the program and program management. Since this survey result was from a program manager, at least two plausible explanations may exist. Either the program manager is answering the survey as how he thinks the program should be run as opposed to how it is actually is run or the processes and structure the program manager has established for the program are not understood well enough by other development team members.

B. RECOMMENDATIONS

Using the survey questionnaires as a guide, program management performance can be improved by evaluating questions where choices selected were not scored as the preferred alternative. Participants taking the survey for the same program over the same timeframe can uncover significant dichotomies when discussing questions where the responses were different.

1. Top-Level Evaluation Sections

Software engineering is not a static discipline. New techniques and improved strategies are constantly being developed. Further re-evaluation of the lower factors in each of the top-level factor sections can serve to refine the basis for evaluating the quality of software program management.

While the QMM score can give the program manager a view of past and present performance, reviewing the questions in factor sections where scores are weaker can provide insight and guidance to the software program manager seeking improvement. The survey is intended to be administered by individuals who understand the elements, motivation, and scoring of the questions and responses in each of the sections. These administering individuals can then provide one-on-one guidance and further explanation to the software program manager throughout the process. Additionally, survey administrators should interview the survey participants to uncover any pre-bias or misperceptions that may exist and influence the survey results.

2. Survey

As new techniques and improved strategies are developed, the questionnaires must be continually refined to assure that higher scores relate to higher software management performance. Immediate future work should focus on refining the questions in each of the questionnaires to better reflect desired outcomes of software programs. This can be accomplished in three ways.

The first way is to improve the wording of existing questions to improve the clarification of concepts and to eliminate wording that could imply a preferred response. If the appearance of response choices is neutral there is less likely a temptation of the survey participant to, consciously or subconsciously, choose the implied correct response rather than the appropriate response reflecting current conditions in the program.

Secondarily, survey improvement may be attained from formulation of replacement questions. The attempt would be to adjust focus on the more important concepts that determine software management quality.

Finally, refining the point values of individual responses can improve the survey. Responses for the more important concepts must be reflected with higher point values than responses given for questions more marginal in determining software management quality.

Based on feedback from survey participants, the current length of the survey is appropriate for coverage of the material important in evaluating software management. However, any increase in length of the survey was viewed as a negative and would discourage participation. Therefore, the emphasis for improvement in the questionnaires must be from refinement and replacement of current questions.

3. Metric Scoring

This thesis provides an informal verification and validation, evaluating only three programs for a QMM score. All three programs were Department of Defense systems. A greater number of software program managers and key team members, in addition to a greater variety of software programs, need to be evaluated to establish a statistically significant correlation of the QMM score to overall software program success. The process is iterative and may necessitate adjustment in scoring the metric to better correlate with software program performance. Particular attention should be noted for programs of various sizes and types. Metric scoring formulation may require different coefficients based on whether the software development is commercial or government. Metric scoring may also require different coefficients based on the size or complexity of software developments. These conclusions can only be attained after significant numbers of surveys are conducted and their results evaluated for statistically significant trends.

As additional surveys are conducted, the collective understanding of what constitutes the quality of software program management will continue to grow. Applying measurement to the quality of software management will lead to improvements of program managers and the likelihood of the success and quality of their software programs.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A
QMM SUMMARY SHEETS FROM ALL SURVEY PARTICIPANTS

Program: C QMM Summary Score Sheet Date: Nov 99

QMM Scoresheet		Part One		Part Two	Total		Importance		Weighted
Category		Score		Score	Score		Coefficient		Score
Requirements Management	a	48	e	34	82	X	0.92	=	75.73
Est./Planning Management	b	50	f	38	88	X	0.67	=	59.01
People Management	c	48	g	51	99	X	1.86	=	184.61
Risk Management	d	33	h	1	34	X	0.55	=	18.60

QMM SCORE 337.95

Max. QMM score possible	**528.00**
Min. QMM score possible	**-130.86**
QMM percentage score:	**71.15%**

Objective/Subjective view of the overall success of program A on a scale of 0 to 10
(0 being total failure, 10 being perfect program total success)
Survey Participant: A-pm
Success Score: 7

Program: C QMM Summary Score Sheet Date: Nov 99

QMM Scoresheet		**Part One**		**Part Two**	**Total**		**Importance**		**Weighted**
Category		**Score**		**Score**	**Score**		**Coefficient**		**Score**
Requirements Management	**a**	44	**e**	35	79	**X**	**0.92**	=	**72.96**
Est./Planning Management	**b**	43	**f**	26	69	**X**	**0.67**	=	**46.27**
People Management	**c**	54	**g**	45	99	**X**	**1.86**	=	**184.61**
Risk Management	**d**	33	**h**	1	34	**X**	**0.55**	=	**18.60**

QMM SCORE	**322.44**

Max. QMM score possible	**528.00**
Min. QMM score possible	**-130.86**
QMM percentage score:	**68.80%**

Objective/Subjective view of the overall success of program A on a scale of 0 to 10
(0 being total failure, 10 being perfect program total success)
Survey Participant: A-1
Success Score: 7

Program: C

QMM Summary Score Sheet

Date: Nov 99

QMM Scoresheet Category		Part One Score		Part Two Score	Total Score		Importance Coefficient		Weighted Score
Requirements Management	a	42	e	39	81	X	0.92	=	74.81
Est./Planning Management	b	57	f	36	93	X	0.67	=	62.36
People Management	c	58	g	50	108	X	1.86	=	201.39
Risk Management	d	44	h	43	87	X	0.55	=	47.59

QMM SCORE	**386.15**

Max. QMM score possible	**528.00**
Min. QMM score possible	**-130.86**
QMM percentage score:	**78.47%**

Objective/Subjective view of the overall success of program A on a scale of 0 to 10
(0 being total failure, 10 being perfect program total success)
Survey Participant: B-pm
Success Score: 7

Program: C QMM Summary Score Sheet Date: Nov 99

QMM Scoresheet		Part One		Part Two	Total		Importance		Weighted
Category		**Score**		**Score**	**Score**		**Coefficient**		**Score**
Requirements Management	a	29	e	13	42	X	0.92	=	38.79
Est./Planning Management	b	19	f	-13	6	X	0.67	=	4.02
People Management	c	21	g	5	26	X	1.86	=	48.48
Risk Management	d	17	h	9	26	X	0.55	=	14.22

QMM SCORE	**105.52**

Max. QMM score possible	**528.00**
Min. QMM score possible	**-130.86**
QMM percentage score:	**35.88%**

Objective/Subjective view of the overall success of program A on a scale of 0 to 10
(0 being total failure, 10 being perfect program total success)
Survey Participant: B-1
Success Score: 7

Program: C QMM Summary Score Sheet Date: Nov 99

QMM Scoresheet		Part One		Part Two	Total		Importance		Weighted
Category		Score		Score	Score		Coefficient		Score
Requirements Management	a	16	e	6	22	X	0.92	=	20.32
Est./Planning Management	b	21	f	-16	5	X	0.67	=	3.35
People Management	c	25	g	-10	15	X	1.86	=	27.97
Risk Management	d	6	h	-15	-9	X	0.55	=	-4.92

QMM SCORE 46.72

Max. QMM score possible	**528.00**
Min. QMM score possible	**-130.86**
QMM percentage score:	**26.95%**

Objective/Subjective view of the overall success of program A on a scale of 0 to 10
(0 being total failure, 10 being perfect program total success)
Survey Participant: B-2
Success Score: 7

Program: C QMM Summary Score Sheet Date: Nov 99

QMM Scoresheet Category		Part One Score		Part Two Score	Total Score		Importance Coefficient		Weighted Score
Requirements Management	a	23	e	1	24	X	0.92	=	22.16
Est./Planning Management	b	11	f	-20	-9	X	0.67	=	-6.04
People Management	c	52	g	48	100	X	1.86	=	186.47
Risk Management	d	12	h	-21	-9	X	0.55	=	-4.92

QMM SCORE 197.68

Max. QMM score possible	**528.00**
Min. QMM score possible	**-130.86**
QMM percentage score:	**49.86%**

Objective/Subjective view of the overall success of program A on a scale of 0 to 10
(0 being total failure, 10 being perfect program total success)
Survey Participant: C-pm
Success Score: 7

Program: C QMM Summary Score Sheet Date: Nov 99

QMM Scoresheet		Part One		Part Two	Total		Importance		Weighted
Category		Score		Score	Score		Coefficient		Score
Requirements Management	a	29	e	7	36	X	0.92	=	33.25
Est./Planning Management	b	18	f	5	23	X	0.67	=	15.42
People Management	c	37	g	43	80	X	1.86	=	149.18
Risk Management	d	7	h	-23	-16	X	0.55	=	-8.75

QMM SCORE	189.09

Max. QMM score possible	528.00
Min. QMM score possible	-130.86
QMM percentage score:	48.56%

Objective/Subjective view of the overall success of program A on a scale of 0 to 10
(0 being total failure, 10 being perfect program total success)
Survey Participant: C-1
Success Score: 7

APPENDIX B
COMPLETE SURVEYS FROM THE PROGRAM MANAGERS

Pair choice section ONE: (Requirements Management) choose most applicable term of the two for each row (page 1 of 2):

formal requirement list	X	informal requirement list	
written requirements	X	oral requirements	
requirements informal, but recorded	X	requirements not recorded	
requirements as part of an SRS (or other formal repository)	X	requirements informally recorded	
requirements taken as is from customer		look to reformulate, interview in-depth, or otherwise re-validate	X
only one development strategy used	X	strategies not consistent, used at different times	
stakeholders as part of requirements development		stakeholders approving requirements after formulated by development team	X
requirements are testable	X	requirements have no test plans	
informal test plan or no test plan		formal test plan	X
test team involved with requirements	X	no test team input or plans during requirements development	
only a percentage of requirements present in baseline		baseline must contain all requirements	X
requirements documentation has hierarchical structure		all requirements must be implemented	X
requirements have listed responsible party		requirements origin not important	X
requirements documentation have versions	X	no requirements history	
requirements have specific attribute values		requirements all rank evenly	X
funding controls requirements definition		requirements definition controls funding	X
requirements are top down	X	requirements are bottom up	
users/stakeholders are identified and interviewed (market survey)		no special consideration to identify users/stakeholders	X
each requirement has a singular concept		some requirements are compound statements	X
requirements definition minimized when funding short		program scope may reduce, but requirements definition completed	X
requirements extraction has formal process	X	requirements extraction ad hoc	
change procedures formal	X	change procedures ad hoc	
users/stakeholders somehow involved in requirements definition	X	program team only involved in requirements definition	
management sets requirements for developers		developers at least partially involved in setting requirements	X
requirements changed at least once since baseline established prior to new version	X	requirements in baseline has not changed prior to new version or upgrade	
no ranking of requirements	X	requirements have priorities assigned	
use-case diagrams (or other models or scenario developments)		no models used for requirements extraction	X
requirements changes informal		requirements changes formal	X
plan to "freeze" requirements as some designated milestone	X	no provision for "freezing" requirements	
requirements must be traceable	X	origin of requirements not important	
requirements must be testable	X	system developed must be testable	
test plans to determine requirements implemented	X	no test plans needed for requirements verification	
requirements have priorities in implementation		all requirements must be implemented	X
some requirements have multiple statements or ideas	X	one idea, one statement per requirement	

Requirements Management (page 1 of 2) score 36

Program Name A

Date Nov99

ANSWER THIS BLOCK OF QUESTIONS ONLY IF A SEQUENTIAL OR WATERFALL APPROACH IS USED FOR DEVELOPMENT (Requirements page 2 of 2)			
requirements first, then initial development work	X	initial development work then requirements	
requirements documentation driving development	X	requirements documentation developed in parallel/after development	
user feedback considered during development		after development starts, user feedback serves as input to new work	X
change management procedures used strictly	X	change management procedures as guidance only	
design decisions prior to or in parallel to requirements development	X	design decisions only after approved requirements stabilized	
requirements summarized what we have developed		requirements are the blueprint for development	X
length of time for requirements work greater than development work		length of time for requirements work less than development work	X
requirements have design detail	X	no design detail in requirements	
requirements creep to be avoided		requirements creep o.k., but need to be controlled	X
freeze requirements at some point	X	requirements are fluid throughout development	
formal change procedure	X	informal change procedure	
change management plan	X	no change management plan	
requirements ambiguity always present to some extent	X	requirements ambiguity unacceptable at any level	
testing considered up front during requirements determination		testing considered down the line during development	X
requirements development team members different from implementation		those working on requirements, work on implementation	X
start implementation as early as possible to help define requirements		requirements must be defined prior to any implementation work	X
ANSWER THIS BLOCK OF QUESTIONS ONLY IF A PROTOTYPING, THROWAWAY, SYNCHRONIZE & STABILIZE, OR OTHER STRATEGY USED			
develop prototype, then determine requirements		determine requirements prior to any development work	
requirements testing done after each iteration		no testing	
individual changes as necessary		only block changes made	
development team decides on changes after iteration		users involved with changes	
changes based on feedback only from user for correction of problems		changes to upgrade system and correct problems	
funding controls changes and change procedures		changes control funding	
requirements documentation finalized prior to development		requirements fluid throughout development (only freeze at end)	
requirements test plans completed prior to development		requirements test plans completed after development	
requirements first, then initial development work		initial development work then requirements	
use development effort to learn more about requirements		define all requirements prior to coding anything	
requirements ambiguity always present to some extent		requirements ambiguity unacceptable at any level	
requirements have design detail		no design detail in requirements	
user feedback considered during development		after development starts, user feedback serves as input to new work	
get something to users as soon as possible for evaluation		make sure it is complete before releasing	
management dictates requirements		developm't team visually represent requirements through rapid prototyping	
new requirements allowed after initial requirements defined		new requirements not allowed	

Requirements Management (pg 2 of 2) score 12 **+ pg 1 score** 36 **= TOTAL SCORE** 48 **Enter on QMM scoresheet blk a.**

Program Name A

Date Nov99

Pair choice section TWO: (Estimation/Planning Management) choose most applicable term of the two for each row (page 1 of 2):

At least one estimation method used in program	X	No estimates	
Formal derivation of product metric for estimation of size	X	Ad hoc size estimation	
Ad hoc process evaluation		Formal derivation of at least one process metric	X
Develop work breakdown structure (WBS)	X	Assign work as needs arise	
Estimates are developed to fulfill a data call only		Use estimates to plan program	X
Use estimates to sell program only		Estimates are useful to the project team for planning purposes	X
Resource evaluations made for program	X	No resource evaluation for planning	
Use both bottom up & top down for estimate, use one stakeholders like	X	Use both bottom up & top down and evaluate significant differences	
Estimates made and not updated		Estimates updated throughout program	X
Resources estimations used to adjust product size estimate		Estimations made irregardless of resources available	X
Estimations made to fit budget		Budget made from estimations	X
Estimations compromised to get program		Rather risk loss of program than compromise confident estimations	X
Cycle time estimations		No cycle time estimations	X
Event count estimations	X	No event count estimations	
Lines of code (LOC) estimation	X	No LOC estimation	
Function Point (FP) estimation		No FP estimation	X
Estimates by algorithmic methods	X	Estimates by analogy	
Expert judgement for estimation	X	Ad hoc estimates	
Estimates by algorithmic methods	X	Ad hoc estimates	
Expert judgement for estimates		Estimates by analogy	X
Ad hoc estimates		Estimates by analogy	X
Bottom up estimates	X	Expert judgement	
Top down estimates		Expert judgement	X
Ad hoc estimates	X	Any other estimate process	
Fuzzy logic estimating method	X	No formal estimation methodology	
WBS development from estimates		WBS development in parallel or prior to estimation completion	X
Critical path of program determined	X	Tasks developed but no path is identified	
Estimators are program team members	X	Estimators are outside program team	
Management only on estimations		All team members involved in estimation process	X
Estimates updated at reviews	X	No updates of estimates	
Estimates updated at reviews	X	Estimates constantly updates (in between reviews, too)	
Estimate procedures stay the same	X	Estimate procedures change	
Stakeholders are part of estimation process		Stakeholders brief estimations after completion	X
Estimates are used beyond initial selling of program	X	Estimates are one time events, used for a specific purpose once	
WBS has objective measure of completeness		Important to have WBS as guide, not rigid implementation	X

Estimation/Planning Management page 1 of 2 score 25

Program Name A

Date Nov99

Pair choice section TWO: (Estimation/Planning Management) choose most applicable term of the two for each row (page 2 of 2):

Life cycle estimates		Estimates for program initiation only	X
System upgrades (SCR) software change requests estimated individually	X	Systems upgrades estimated as whole	
Estimates for on-going resources needed to maintain s/w	X	Estimates for maintenance not done	
Informal re-estimates during development		Formal re-estimates at pre-defined milestones	X
Formal re-estimates when amendment changing the system is introduced	X	Informal re-estimates when amendment changing the system	
Person in-charge of estimation walks in a managers office to get an opinion		Meeting(s) organized for purpose of performing cost estimations	X
Factor analysis prior to commencement of program	X	None done	
Change control procedures set in place	X	No set procedures	
Elapsed time and actual work time estimates		one or the other or neither	X
No schedule created		Schedule created	X
Schedule not updated		Schedule updated	X
Schedule followed	X	Schedule not followed	
Tasks identification arises as program progresses	X	Detailed level tasks identified prior to program initiation	
Scope of program understood by all	X	Scope not explicitly defined	
Quality factors and criteria identified	X	No explicit quality factors defined	
No project tracking tools used		Project tracking tools used	X
CSCIs identified and tasked	X	CSCIs not explicitly identified	
Expectations are managed via estimations	X	Estimations are made to fit preconceived expectations	
No cost schedule developed		Cost schedule developed	X
No resource schedule developed		Resource schedule developed	X
Team members, management know at any time if in budget & schedule		Exact budget & schedule status somewhat unclear to at least some	X
Individual program phases are estimated	X	Only top level program estimated	
Stakeholders/users emphasis understood- quick to field or all complete		Program management sets delivery tradeoffs without outside input	X
Testing planned with initial program planning	X	Testing no in initial planning	
Documentation not considered in initial planning		Documentation part of initial planning	X
Hardware considered in estimations	X	Software only considered	
No formal schedule/cost tracking		Formal procedures established for tracking cost and schedule	X
Earned value set up		Earned value not used	X
Estimations omit documentation planning	X	Documentation in estimates	
Training omitted in estimates		Training part of estimates	X
Earned value set up, but not tracked	X	Earned value tracked	
Detailed planning done with incomplete set of requirements	X	Detailed planning done with detailed set of requirements	
Complete infrastructure support mechanism understood for estimations		No consideration of infrastructure done for estimations	X
Team possibilities considered for planning of program	X	No consideration for outside teaming possibilities	
Work Breakdown Structure (WBS) set up	X	No WBS completed	

Estimation/Planning Management pg 2 of 2 score 25 **+ pg 1 of 2 score** 25 **= TOTAL SCORE** 50 **Enter QMM scoresheet blk b.**

Program Name A

Date Nov99

Pair choice section THREE: (People Management) choose most applicable term of the two for each row (page 1 of 2):

Human Resources

Program team members have clearly defined, segmented roles	X	Work responsibilities are shared	
Formal team building procedures are used		No formal team building emphasized	X
Program manager flexible regarding work hours	X	Program manager maintains strict standards for work hours	
Big picture conveyed to all team members by program management	X	Program management focuses on the partitioned tasks with team	
People issues dealt with primarily through indirect methods (email, memo, etc)		People issues dealt with primarily through direct methods (face-to-face)	X
Training is required and planned on a regular basis		Training is ad hoc	X
Each team member is educated on and understands overall program and their roles		Team members only know their respective areas	X
Consideration for team members' career goals are reflected in assignments	X	Team members must adapt to tasks that are assigned	
Team members assignments and responsibilities are mostly dictated by PM		Assignments and responsibilities are discussed and agreed upon with PM	X
Management leads in problem solving		Management facilitates and lets team lead in problem solving	X
Management welcomes problems as challenges and opportunities	X	Management views problems as obstacles and grounds for punishment	
Team members participate in performance evaluations of peers		Personnel evaluations are strictly PM responsibility	X
Management reinforcement feedback sparse and inconsistent, if any	X	Management provides timely reinforcement feedback for positive behaviors	
Management provides basic needs of office facilities fairly well	X	Office facilities are a drawback to working in the program	
Working conditions are fairly comfortable, time off policy fairly good	X	Working conditions and time off policy is inconsistent and difficult at times	

Communication:

Communications primarily written (email)		Communications primarily verbal (face-to-face)	X
Detailed instructions: oral presentation, follow-up email	X	Email only	
Formal communication protocol		Informal communications	X
External vertical communications restricted	X	External vertical communication allowed	
Coders notebook, weekly accomplishment reports required	X	Not required	
User-coder relationship established, encouraged, and mediated		User-coder interaction minimized	X
Meetings structured to minimize wasted time	X	Meetings unstructured and open ended	
Meetings have agenda, objectives, and conclude with action items	X	Meeting agenda fluid and open ended	
Program management and coder communication face to face	X	Program management and coder communication primarily email	
Program team updated regularly regarding organizational & program status		Meetings infrequently scheduled	X
Open communications is encouraged	X	Communication through chain of command only is encouraged	
Program manager accessible for discussions	X	Program manager difficult to get an appointment to see	
Program management (PM) is viewed as separate from team		PM mixes with team frequently	X
Management regularly holds team meetings	X	Meetings are sporadic	
Meetings are structured with definite goals and objectives	X	Meetings are informal	
Program management is generally easy to reach and talk to	X	PM is usually hard to get a hold of and difficult to talk to	
Team-program manager relationship adult-adult	X	Team-program management relationship parent-child	
Schedules are spontaneous and poorly communicated		Schedules must be fixed and rigidly followed and formally reported	X
Work is seen as complex processes involving team working together	X	Work broken into pieces with minimal team member interaction	
Action items often is poorly disseminated and usually not followed through		Action items communicated and followed through thoroughly	X
Team members require frequent clarifications by PM for assigned tasks		Team members rarely require clarifications by PM for assigned tasks	X

Program Name A

Date Nov99

Pair choice section THREE: (People Management) choose most applicable term of the two for each row (page 2 of 2):

Leadership:

Long range organizational vision		Short term program and immediate work focus	X
Lead through personal attention to others		Action-oriented leadership approach	X
Run as much of the organization as possible		Let team make decisions as much as possible	X
Direct and domineering style	X	Encourage independence in others	
Traditional leaders respect hierarchy		Do what needs to be done	X
Win cooperation rather than demand it	X	Tough-minded with others	
Act strongly and forcefully in the field of ideas	X	Prefer to lead other independent types while seeking autonomy for self	
Consults with team members to find solutions to problems	X	Consults team members to get validation of PM's predetermined solutions	
Keep people well informed	X	Only as much knowledge as necessary for their work	
Make things happen by focusing on the immediate problems	X	Long range focus and de-emphasize current problems	
Manage others loosely and prefer minimal supervision		Follow traditional procedures and rules conscientiously	X
Leadership, management decisions exclusively by program management		Program management makes decisions but gets inputs from team	X
team-program manager relationship adult-adult	X	Team-program management relationship parent-child	
Program management makes decisions but gets inputs from team	X	All program team members responsible for program decisions	
When a problem arises: management takes over to solve it	X	Management lets the team solve the problems	
Leadership is do as I say, not do as I do		Leadership by example	X
Program expectation not influenced by PM		Program expectation managed by PM	X
PM gives freedom to team, but does has no mentoring for members (abdication)	X	PM empowers teams by mentoring members to be leaders	
Program management waits and sees what happens then plans		Management plans far in advance	X
Program management is constantly reacting to emergencies	X	Management is one step ahead of problems	
Facilitative approach to solving problems		Take charge readily and often	X
Program management is complex, takes much time to understand	X	Management is simple, easy to figure out	
Program management prefers to plunge right in	X	Takes time to separate things to be done and order of doing them	
Program management reacts spur of the moment	X	Methodically follows plans	

Technical Competency of the program manager:

PM has technical experience particular to the particular s/w program	X	PM relies on team members solely	
PM participates in technical reviews	X	PM only in non-technical reviews	
PM participates in making technical decisions when problems arise	X	PM delegates technical questions	
PM does not get involved discussing technical options		PM contributes to technical options being discussed	X
PM does not review technical options and decisions		PM reviews technical options and decisions	X
PM actively attempts to keep up-to-date with current technology and standards	X	PM is removed from cutting edge technology issues	
PM receives technical periodicals and occasionally references applicable articles	X	PM doesn't read periodicals nor references current articles to team	
PM doesn't have technical background (or education)		PM has technical background (or education)	X
Team members avoid PM when they need technical advice		Team members generally consider talking to PM regarding technical issues	X

HR 9 **+ Comm.** 17 **+ Leadership** 13 **+ Tech. Competency** 9 **= People Mgmt. score** 48 **Enter on blk c.**

Program Name A

Date Nov99

Pair choice section Four: (Risk Management (RM)) choose most applicable term of the two for each row (page 1 of 2):

RM is formal and documented		RM is informal, if at all	X
A risk management plan exists		No risk management plan is developed	X
RM is more of a data call than a useful document		RM drives decisions on the program	X
RM is done prior to the program beginning		RM is done prior and during program execution	X
RM is only done during the program execution	X	RM is done prior and during program execution	
Risks are generalized through the whole program	X	Risks are categorized	
Risk management is done internally, only	X	An outside organization also contributes to the RM process	
Risk is a management function		Risk is a program team function	X
Risk are precisely articulated		Risks are generalized, if at all	X
Each risk has a consequence		Consequences are generalized, if at all	X
a mitigation strategy is completed for each risk		Mitigation strategy is generalized, if at all	X
Contingency plans are developed for a RM plan		Contingency plans are ad hoc as problems arise in the program	X
Risks are anticipated		if problems arise, management will deal with it	X
the program doesn't have any risk		Programs that do not have risk, have problems	X
Risk management is automated		Risk management may use tools, but depend on human input	X
Risks are assigned probabilities		Probabilities are not relevant for RM	X
all risks are potential problems, relative priorities for risks are not useful	X	Risks are weighed relative to other program risks and thus prioritized	
Risk management information is only shared internally	X	Risk management information is shared with all stakeholders	
Risk analysis uses ordinal rankings	X	Risk analysis uses actual measurements with a mathematical model	
Regret analysis used		No regret analysis done	X
Attach probabilities to future events		No probabilities associated with future events	X
Assessing risks with mechanical methods		Risks should be compared to other risks and sorted	X
Risk status tracked		Not tracked	X
Technical risks examined	X	No technical risks examined	
Process risks examined	X	No process risks examined	
Product risks examined	X	No product risks examined	
Stakeholder/user risks examined	X	No examination of stakeholder/user risks	
Checklists used to identify risks		No checklists used	X
Risks are tracked		No tracking or monitoring of risks	X
Each risk has an impact	X	No impact analysis of risk	
Each risk has a mitigation plan	X	No individual risk mitigation	
Risks monitored by priority		No special attention to track higher priority risks	X
Risk assessment is formalized		No formal risk assessment	X
Risk control is formalized		No formal risk control	X
Integration risks not considered		Integration risks examined	X

Risk Management page 1 of 2 score 13

Program Name_____A_________

Date_Nov99

Pair choice section Four: (Risk Management (RM)) choose most applicable term of the two for each row (page 2 of 2):

Risks to cost	X	No cost risks examined	
Unforeseen risks have occurred in program	X	Any risk that came up had been identified previously	
Personnel risks examined	X	No personnel risks examined	
Estimation risks examined	X	No estimation risks examined	
Planning risks examined	X	No planning risks examined	
Requirements risks examined	X	No requirements risks examined	
Resource risks examined	X	No resource risks examined	
Risk management plan updated regularly		No regular risk management plan updates	X
Risks charted		Risks not charted	X
Performance risks examined	X	Performance risks not examined	
Program management self risks examined	X	No program management risks examined	
Risk from program constraints examined	X	No program constraint risks examined	
Each category of risks are prioritized		No prioritization	X
Each category of risks are evaluated for impact	X	No impact analysis performed	
Each category of risks have control strategy	X	No control strategy	
Documentation risks examined	X	No documentation risks examined	
Regret matrix tracked		No regret matrix or not tracked	X
Communication of risk activities are facilitated		No facilitation or promotion of communication of risk activities	X
Taxonomy-based questionnaire used to identify risks		Taxonomy-based questionnaire not used	X
Associated hardware risks examined	X	No consideration for hardware risks	
Integration risks examined	X	Integration risks not examined	
Communication risks examined	X	Communication risks not examined	
Leadership risks examined	X	Leadership risks not considered	
Risk avoidance considered for certain risks	X	Risk avoidance not considered for risks	
Risk documentation forms used		No risk documentation forms used	X
Dependency risks examined	X	No dependency risks examined	
Alternatives like risk avoidance considered for high risk items	X	No consideration of risk avoidance	
Documented risk statements use a condition-consequence type format		Condition-consequence of risk statements not clearly defined	X
no assignment of ownership of risk mitigation action	X	Each risk mitigation action is assigned to an individual for resolution	
Calculation of risk exposure made (probability X loss, for each risk)		No risk exposure calculations	X
Oral communication of risks only	X	Risks written in a way that communicates nature and status of factors	
Triggers used to quantify risk conditions present		Risk conditions present are all subjective	X
Risk "czar" in program for monitoring risks		No special positions/responsibilities for risk monitoring	X
Post-program review completed (scheduled) for unanticipated problems ID		No post-program reviews completed or scheduled	X
No schedule risks examined		Risks to schedule investigated	X

Risk Management pg 2 of 2 score 20 **+ pg 1 of 2 score** 13 **= TOTAL SCORE** 33 **Enter on QMM scoresheet blk d.**

Program Name A

Date Nov99

Program Name: A

YES-NO-N/A Questionnaire

Date: November 1999

No.	Requirements Management Questionnaire	Yes	No	N/A	
1	PM chose to have a formal requirements list	X			
2	Requirements recorded in some way	X			
3	Written requirements were part of some formal document	X			
4	Written requirements were informal		X		
5	At least some requirements were oral only		X		
6	All stakeholders were identified		X		
7	All stakeholders participated in the requirements extraction		X		
8	Some stakeholders participated in the requirements extraction	X			
9	Management extracted requirements, no stakeholder involvement		X		
10	Management passed requirements to development team	X			
11	Stakeholders not involved in Management extraction, but approves		X		
12	Management gets inputs from stakeholders, then develops requirments	X			
13	Developers work informally with users to arrive at requirements		X		
14	Same as 13, but management oversees and formalizes		X		
	If a waterfall or sequential development strategy:				
15	All requirements complete before design		X		
16	Some requirements left incomplete prior to design	X			
17	Requirements informal prior to design effort		X		
18	Requirements serve as input	X			
19	Length of time for requirements work greater than development work		X		
20	Requirements developed in parallel to design		X		
OR	***If a prototype, throwaway, or other development strategy:***				
15	Learn about requirements through`development efforts				
16	No coding until all requirements are defined				
17	Requirements formal prior to design effort				
18	Requirements serve as output				
19	Requirements definition work in parallel to development efforts				
20	Requirements developed in parallel to design				
21	Are requirements frozen at some phase	X			
22	Change management exists	X			
23	Change management is formal	X			
24	Project strategy is consistent throughout development	X			
25	Requirements are updated	X			
26	Configuration Management (CM) exists	X			
27	CM is formal	X			
28	Requirements are testable	X			
29	Requirements testing considered/implemented during extraction	X			
30	Requirements testing plan exists	X			
31	Requirements testing is formal	X			
32	All requirements have priorities		X		
33	All requirements must be implemented	X			
34	Requirements are tested	X			
35	All requirements are equally important	X			
36	At least some requirements have priorities		X		
37	All requirements are traceable	X			
38	Traceability not important		X		
39	Each requirement has an author	X			
40	Who authored requirement is not important	X			
41	Initial set of requirements to be implemented, no requirements creep		X		
42	Structured and tracked changes to requirements only		X		
43	Change is inevitable, changes allowed at all times		X		
44	Change is inevitable, but changes limited	X			
45	Requirements control funding	X			
46	Requirements history kept	X			
47	Baseline established for requirements at some point prior to develop	X			Total
	TOTAL SCORING	33	1	0	34

Enter total score on QMM score sheet block e.

Program Name: A

YES-NO-N/A Questionnaire

Date: November 1999

No.	Estimation/Planning Questionnaire	Yes	No	N/A	
1	A volume product metric used (LOC, # of files, # of screens, pages of doc)	X			
2	Measure used for various product elements (modules, components, CSCI)	X			
3	Product measures made by phase (amt at implementation, LOC changed at unit test)		X		
4	Other product attributes measured (FP, throughput, mem cap, cyclomatic complexity)	X			
5	Product metrics tracked and updated throughout program execution	X			
6	Event count process metric used (# defects in test, reqmt changes, milestones met)		X		
7	Time measure process metric used (cycle time)		X		
8	Process metrics tracked and updated throughout program execution		X		
9	Program cost estimations made from product or process metrics	X			
10	Program cost estimations tracked and updated to reflect progress/changes	X			
11	Factor analysis performed on program		X		
12	Program's primary purpose, including major functions and deliverables known	X			
13	Work breakdown structure developed	X			
14	Task estimated with realistic expectations of productivity probabilities	X			
15	Schedules developed based on realistic expectations	X			
16	Schedules tracked and updated based on new information	X			
17	Detailed activity lists used for clearly defined completed/not completed tasks		X		
18	Quality assurance plan or similar to aid in detecting defects early in program		X		
19	COCOMO estimates performed	X			
20	CSCI clearly defined and tasked	X			
21	Estimates completed ad hoc		X		
22	Gantt charts used and updated	X			
23	Resource estimations (working hrs, job categories, task activities) done	X			
24	Earned value established		X		
25	Earned value tracked throughout program		X		
26	Quality expectations established for product with users and stakeholders	X			
27	Critical path for program tasks developed and tracked	X			
28	Meaure of effectiveness (MOE) or Figure of merit established and tracked		X		
29	Estimates are updated routinely	X			
30	Schedules are updated routinely	X			
31	Estimations are made by program management (top-down)	X			
32	Estimations are made by program team members (bottom-up)	X			
33	Automated program tracking used		X		
34	PM usually thorough in tracking and reporting schedules and financials	X			
35	WBS developed only as data call		X		
36	Earned value used to track program progress		X		
37	PM insists on prioritizing work reduction as schedule/funding compromised by stakeholders	X			
38	Estimations are done using both top down and bottoms up approaches	X			
39	All program team members involved in planning process	X			
40	Hardware also considered in estimation process	X			
41	Program history compiled		X		
42	System upgrades (SCR) software changes requests estimated individually	X			
43	Management duties apart of each team member's responsibilities		X		
44	PM dictates schedules to program team		X		
45	Code reviews planned in schedule	X			
46	Defined tangible milestones established for program tasks	X			
47	Test planning done at the start of the program	X			
48	Estimations are completed by those performing the tasks	X			
49	Sensitivity analysis performed for program choices		X		
50	Software deployment planning completed	X			Total
	TOTAL SCORING	43	-5	0	38

Enter total score on QMM score sheet block f.

Program Name: A

YES-NO-N/A Questionnaire

Date: November 1999

No.	People Management Questionnaire	Yes	No	N/A	
1	PM is accessable in person by each team member	X			
2	PM is accessable via email (memo, letter) by each team member	X			
3	PM is accessable via phone by each team member	X			
4	PM not only considers a person's suitability, not also desire to be on a team	X			
5	PM consults with each team member regarding their career goals	X			
6	PM regularly holds meetings to inform team of program progress	X			
7	PM solicits opinions from team members before making decisions	X			
8	PM lets teams make decisions affecting their work		X		
9	PM frequently makes decisions without any consultation with members		X		
10	PM understands the technology/language of the program	X			
11	PM is able to communicate with other the technical issues in the program	X			
12	PM prioritizes problems or conflicts within the program	X			
13	PM assists team members in developing/advising of career path		X		
14	PM empowers program members to recommend hiring new team members	X			
15	PM empowers program members to recommend firings of other members	X			
16	PM specifically assigns work to each program member		X		
17	PM sets communication protocol	X			
18	PM allows unrestricted communications	X			
19	PM encourages or requires training for each individual	X			
20	PM takes control in difficult/ problem areas	X			
21	PM looks ahead to new programs, new upgrades of existing program	X			
22	PM maintains regular communications with all stakeholders	X			
23	PM maintains regular communications with users		X		
24	PM encourages program team communication with users	X			
25	PM encourages program team communication with stakeholders	X			
26	PM facilitates horizontal communication within program	X			
27	PM facilitates communication during integration	X			
28	PM holds meetings without clear objectives		X		
29	PM must approve all decisions within the program		X		
30	PM must approve all interactions with stakeholders		X		
31	PM must approve all interactions with users		X		
32	PM makes all presentations to stakeholders/users		X		
33	PM is considered "flexible" in terms of program members personal issues	X			
34	PM, at least occasionally, schedules/promotes outside work team activities	X			
35	PM is readily willing to listen to program problems and complaints	X			
36	PM takes action to resolve program problems and complaints	X			
37	PM is generally respected by stakeholders, users, and organization	X			
38	PM sometimes fails to grasp important technical issues in program		X		
39	PM recruits program team members from outside organization		X		
40	PM participates in technical reviews	X			
41	Program personnel have clearly defined specific tasks	X			
42	Although individual's tasks are specific, each exposed to the "bigger picture"	X			
43	PM has clearly defined his/her expectations for each individual	X			
44	PM delegation of duties is usually seemless in execution	X			
45	PM acts as facilitator to solving personnel conflicts	X			
46	PM attempts to motivate individuals on the program team	X			
47	PM clearly separates technical from managerial roles for individuals		X		
48	PM directs how he/she expects the task to be accomplished		X		
49	PM directs what needs to be done, but does not direct how	X			
50	PM attempts to spotlight individuals in the program for positive exposure	X			Total
	TOTAL SCORING	44	7	0	51

Enter total score on QMM score sheet block g.

Program Name: A

YES-NO-N/A Questionnaire

Date: November 1999

No.	Risk Management Questionnaire	Yes	No	N/A	
1	Risk Management (RM) is specifically an activity in the program		X		
2	RM is formal and documented		X		
3	A specific RM plan exists		X		
4	RM is required in the program, but not used during the program		X		
5	RM is done prior to the program execution		X		
6	RM is done by an outside entity to the development		X		
7	RM is done internally only	X			
8	RM is both internally performed and externally assessed		X		
9	RM planning occurs during or after major milestones in the program	X			
10	Risk Assessment is only a management function	X			
11	RM is informal or non existent	X			
12	There is a RM plan, but it is not updated or tracked		X		
13	Risks are only generalized	X			
14	Each risk is delineated		X		
15	Each risk has a consequence		X		
16	Each risk has a likelihood rating of some sort		X		
17	Each risk has a mitigation strategy		X		
18	Risk Management is automated		X		
19	Risks are tracked		X		
21	Regret analysis performed		X		
22	RM drives decisions in the program		X		
23	Risks have probabilities		X		
24	Risk Management is ad hoc	X			
25	RM information is shared with all stakeholders (as appropriate)	X			
26	Risks are weighed relative to other program risks	X			
27	Risk Assessment is a program team activity		X		
28	Risk Assessment done prior to program start		X		
29	Risk Assessment includes personnel risk	X			
30	RM uses tools, but depends on human decisions		X		
31	Risk Assessment includes cost risks	X			
32	Risk Assessment includes schedule risks	X			
33	Risk Assessment includes technology risks	X			
34	Risk Assessment is briefed organization structure above program manager		X		
35	Risk Assessment includes requirements risks	X			
36	Risk Assessment includes user risks (too little involvement of user)	X			
37	Risk Assessment includes documentation risks	X			
38	Risk Assessment includes integration risks	X			
39	Risk Assessment includes interface risks (non-standard)	X			
40	Risk Assessment includes continuing requirements change (feature creep)	X			
41	Risk Assessment includes dependent projects/programs risks	X			
42	Documentation proof exists to demonstrate following risk management plan		X		
43	High risk have measured tracking (high profile status)		X		
44	Organizational history used to search for risks	X			
45	Other organizational checklists used for risk assessment		X		
46	Internal organizational checklists used for risk assessment	X			
47	Risk Assessment information contributed to internal or other database		X		
48	Risk Assessment includes internal organization risks	X			
49	Risk Assessment includes stakeholder risks	X			
50	No risk management needed; program is straightforwarded & understood		X		Total
	TOTAL SCORING	14	-13	0	1

Enter total score on QMM score sheet block h.

Pair choice section ONE: (Requirements Management) choose most applicable term of the two for each row (page 1 of 2):

formal requirement list	X	informal requirement list	
written requirements	X	oral requirements	
requirements informal, but recorded	X	requirements not recorded	
requirements as part of an SRS (or other formal repository)	X	requirements informally recorded	
requirements taken as is from customer	X	look to reformulate, interview in-depth, or otherwise re-validate	
only one development strategy used		strategies not consistent, used at different times	X
stakeholders as part of requirements development		stakeholders approving requirements after formulated by development team	X
requirements are testable	X	requirements have no test plans	
informal test plan or no test plan		formal test plan	X
test team involved with requirements	X	no test team input or plans during requirements development	
only a percentage of requirements present in baseline		baseline must contain all requirements	X
requirements documentation has hierarchical structure		all requirements must be implemented	X
requirements have listed responsible party		requirements origin not important	X
requirements documentation have versions	X	no requirements history	
requirements have specific attribute values		requirements all rank evenly	X
funding controls requirements definition	X	requirements definition controls funding	
requirements are top down	X	requirements are bottom up	
users/stakeholders are identified and interviewed (market survey)	X	no special consideration to identify users/stakeholders	
each requirement has a singular concept		some requirements are compound statements	X
requirements definition minimized when funding short	X	program scope may reduce, but requirements definition completed	
requirements extraction has formal process	X	requirements extraction ad hoc	
change procedures formal	X	change procedures ad hoc	
users/stakeholders somehow involved in requirements definition	X	program team only involved in requirements definition	
management sets requirements for developers		developers at least partially involved in setting requirements	X
requirements changed at least once since baseline established prior to new version	X	requirements in baseline has not changed prior to new version or upgrade	
no ranking of requirements	X	requirements have priorities assigned	
use-case diagrams (or other models or scenario developments)		no models used for requirements extraction	X
requirements changes informal		requirements changes formal	X
plan to "freeze" requirements as some designated milestone	X	no provision for "freezing" requirements	
requirements must be traceable	X	origin of requirements not important	
requirements must be testable	X	system developed must be testable	
test plans to determine requirements implemented	X	no test plans needed for requirements verification	
requirements have priorities in implementation		all requirements must be implemented	X
some requirements have multiple statements or ideas	X	one idea, one statement per requirement	

Requirements Management (page 1 of 2) score 30.

Program Name B

Date Nov99

ANSWER THIS BLOCK OF QUESTIONS ONLY IF A SEQUENTIAL OR WATERFALL APPROACH IS USED FOR DEVELOPMENT ***(Requirements page 2 of 2)***			
requirements first, then initial development work		initial development work then requirements	
requirements documentation driving development		requirements documentation developed in parallel/after development	
user feedback considered during development		after development starts, user feedback serves as input to new work	
change management procedures used strictly		change management procedures as guidance only	
design decisions prior to or in parallel to requirements development		design decisions only after approved requirements stabilized	
requirements summarized what we have developed		requirements are the blueprint for development	
length of time for requirements work greater than development work		length of time for requirements work less than development work	
requirements have design detail		no design detail in requirements	
requirements creep to be avoided		requirements creep o.k., but need to be controlled	
freeze requirements at some point		requirements are fluid throughout development	
formal change procedure		informal change procedure	
change management plan		no change management plan	
requirements ambiguity always present to some extent		requirements ambiguity unacceptable at any level	
testing considered up front during requirements determination		testing considered down the line during development	
requirements development team members different from implementation		those working on requirements, work on implementation	
start implementation as early as possible to help define requirements		requirements must be defined prior to any implementation work	
ANSWER THIS BLOCK OF QUESTIONS ONLY IF A PROTOTYPING, THROWAWAY, SYNCHRONIZE & STABILIZE, OR OTHER STRATEGY USED			
develop prototype, then determine requirements		determine requirements prior to any development work	X
requirements testing done after each iteration	X	no testing	
individual changes as necessary	X	only block changes made	
development team decides on changes after iteration	X	users involved with changes	
changes based on feedback only from user for correction of problems		changes to upgrade system and correct problems	X
funding controls changes and change procedures	X	changes control funding	
requirements documentation finalized prior to development	X	requirements fluid throughout development (only freeze at end)	
requirements test plans completed prior to development		requirements test plans completed after development	X
requirements first, then initial development work	X	initial development work then requirements	
use development effort to learn more about requirements	X	define all requirements prior to coding anything	
requirements ambiguity always present to some extent	X	requirements ambiguity unacceptable at any level	
requirements have design detail	X	no design detail in requirements	
user feedback considered during development	X	after development starts, user feedback serves as input to new work	
get something to users as soon as possible for evaluation	X	make sure it is complete before releasing	
management dictates requirements	X	developm't team visually represent requirements through rapid prototyping	
new requirements allowed after initial requirements defined	X	new requirements not allowed	

Requirements Management (pg 2 of 2) score 12 **+ pg 1 score** 30 **= TOTAL SCORE** 42 **Enter on QMM scoresheet blk a.**

Program Name B

Date Nov99

Pair choice section TWO: (Estimation/Planning Management) choose most applicable term of the two for each row (page 1 of 2):

At least one estimation method used in program	X	No estimates	
Formal derivation of product metric for estimation of size		Ad hoc size estimation	X
Ad hoc process evaluation		Formal derivation of at least one process metric	X
Develop work breakdown structure (WBS)	X	Assign work as needs arise	
Estimates are developed to fulfill a data call only		Use estimates to plan program	X
Use estimates to sell program only		Estimates are useful to the project team for planning purposes	X
Resource evaluations made for program	X	No resource evaluation for planning	
Use both bottom up & top down for estimate, use one stakeholders like		Use both bottom up & top down and evaluate significant differences	X
Estimates made and not updated		Estimates updated throughout program	X
Resources estimations used to adjust product size estimate	X	Estimations made irregardless of resources available	
Estimations made to fit budget		Budget made from estimations	X
Estimations compromised to get program		Rather risk loss of program than compromise confident estimations	X
Cycle time estimations	X	No cycle time estimations	
Event count estimations	X	No event count estimations	
Lines of code (LOC) estimation		No LOC estimation	X
Function Point (FP) estimation		No FP estimation	X
Estimates by algorithmic methods		Estimates by analogy	X
Expert judgement for estimation	X	Ad hoc estimates	
Estimates by algorithmic methods	X	Ad hoc estimates	
Expert judgement for estimates		Estimates by analogy	X
Ad hoc estimates		Estimates by analogy	X
Bottom up estimates	X	Expert judgement	
Top down estimates	X	Expert judgement	
Ad hoc estimates		Any other estimate process	X
Fuzzy logic estimating method		No formal estimation methodology	X
WBS development from estimates		WBS development in parallel or prior to estimation completion	X
Critical path of program determined	X	Tasks developed but no path is identified	
Estimators are program team members	X	Estimators are outside program team	
Management only on estimations		All team members involved in estimation process	X
Estimates updated at reviews	X	No updates of estimates	
Estimates updated at reviews		Estimates constantly updates (in between reviews, too)	X
Estimate procedures stay the same		Estimate procedures change	X
Stakeholders are part of estimation process		Stakeholders briefed on estimations after completion	X
Estimates are used beyond initial selling of program	X	Estimates are one time events, used for a specific purpose once	
WBS has objective measure of completeness	X	Important to have WBS as guide, not rigid implementation	

Estimation/Planning Management page 1 of 2 score 28

Program Name B

Date Nov99

Pair choice section TWO: (Estimation/Planning Management) choose most applicable term of the two for each row (page 2 of 2):

Life cycle estimates	X	Estimates for program initiation only	
System upgrades (SCR) software change requests estimated individually	X	Systems upgrades estimated as whole	
Estimates for on-going resources needed to maintain s/w	X	Estimates for maintenance not done	
Informal re-estimates during development		Formal re-estimates at pre-defined milestones	X
Formal re-estimates when amendment changing the system is introduced	X	Informal re-estimates when amendment changing the system	
Person in-charge of estimation walks in a managers office to get an opinion	X	Meeting(s) organized for purpose of performing cost estimations	
Factor analysis prior to commencement of program		None done	X
Change control procedures set in place	X	No set procedures	
Elapsed time and actual work time estimates	X	One or the other or neither	
No schedule created		Schedule created	X
Schedule not updated		Schedule updated	X
Schedule followed	X	Schedule not followed	
Tasks identification arises as program progresses	X	Detailed level tasks identified prior to program initiation	
Scope of program understood by all	X	Scope not explicitly defined	
Quality factors and criteria identified		No explicit quality factors defined	X
No project tracking tools used		Project tracking tools used	X
CSCIs identified and tasked		CSCIs not explicitly identified	X
Expectations are managed via estimations	X	Estimations are made to fit preconceived expectations	
No cost schedule developed		Cost schedule developed	X
No resource schedule developed		Resource schedule developed	X
Team members, management know at any time if in budget & schedule		Exact budget & schedule status somewhat unclear to at least some	X
Individual program phases are estimated	X	Only top level program estimated	
Stakeholders/users emphasis understood- quick to field or all complete	X	Program management sets delivery tradeoffs without outside input	
Testing planned with initial program planning	X	Testing not in initial planning	
Documentation not considered in initial planning		Documentation part of initial planning	X
Hardware considered in estimations	X	Software only considered	
No formal schedule/cost tracking		Formal procedures established for tracking cost and schedule	X
Earned value set up	X	Earned value not used	
Estimations omit documentation planning		Documentation in estimates	X
Training omitted in estimates		Training part of estimates	X
Earned value set up, but not tracked		Earned value set up and tracked	X
Detailed planning done with incomplete set of requirements		Detailed planning done with detailed set of requirements	X
Complete infrastructure support mechanism understood for estimations	X	No consideration of infrastructure done for estimations	
Team possibilities considered for planning of program	X	No consideration for outside teaming possibilities	
Work Breakdown Structure (WBS) set up	X	No WBS completed	

Estimation/Planning Management pg 2 of 2 score 29 **+ pg 1 of 2 score** 28 **= TOTAL SCORE** 57 **Enter QMM scoresheet blk b.**

Program Name B

Date Nov99

Pair choice section THREE: (People Management) choose most applicable term of the two for each row (page 1 of 2):

Human Resources

Program team members have clearly defined, segmented roles	X	Work responsibilities are shared	
Formal team building procedures are used	X	No formal team building emphasized	
Program manager flexible regarding work hours	X	Program manager maintains strict standards for work hours	
Big picture conveyed to all team members by program management	X	Program management focuses on the partitioned tasks with team	
People issues dealt with primarily through indirect methods (email, memo, etc)		People issues dealt with primarily through direct methods (face-to-face)	X
Training is required and planned on a regular basis	X	Training is ad hoc	
Each team member is educated on and understands overall program and their roles		Team members only know their respective areas	X
Consideration for team members' career goals are reflected in assignments	X	Team members must adapt to tasks that are assigned	
Team members assignments and responsibilities are mostly dictated by PM		Assignments and responsibilities are discussed and agreed upon with PM	X
Management leads in problem solving		Management facilitates and lets team lead in problem solving	X
Management welcomes problems as challenges and opportunities	X	Management views problems as obstacles and grounds for punishment	
Team members participate in performance evaluations of peers	X	Personnel evaluations are strictly PM responsibility	
Management reinforcement feedback sparse and inconsistent, if any		Management provides timely reinforcement feedback for positive behaviors	X
Management provides basic needs of office facilities fairly well	X	Office facilities are a drawback to working in the program	
Working conditions are fairly comfortable, time off policy "flexible"	X	Working conditions and time off policy is inconsistent and difficult at times	

Communication:

Communications primarily written (email, memo, etc.)	X	Communications primarily verbal (face-to-face)	
Detailed instructions: oral presentation, follow-up email	X	Email or memo only	
Formal communication protocol	X	Informal communications	
External vertical communications restricted		External vertical communication allowed	X
Coders notebook, weekly accomplishment reports required	X	Not required	
User-coder relationship established, encouraged, and mediated	X	User-coder interaction minimized	
Meetings structured to minimize wasted time	X	Meetings unstructured and open ended	
Meetings have agenda, objectives, and conclude with action items	X	Meeting agenda fluid and open ended	
Program management and coder communication face to face		Program management and coder communication primarily email	X
Program team updated regularly regarding organizational & program status	X	Meetings infrequently scheduled	
Open communications is encouraged	X	Communication through chain of command only is encouraged	
Program manager accessible for discussions	X	Program manager difficult to get an appointment to see	
Program management (PM) is viewed as separate from team	X	PM mixes with team frequently	
Management regularly holds team meetings		Meetings are sporadic	X
Meetings are structured with definite goals and objectives	X	Meetings are informal	
Program management is generally easy to reach and talk to	X	PM is usually hard to get a hold of and difficult to talk to	
Team-program manager relationship adult-adult	X	Team-program management relationship parent-child	
Schedules are spontaneous and poorly communicated		Schedules must be fixed and rigidly followed and formally reported	X
Work is seen as complex processes involving team working together	X	Work broken into pieces with minimal team member interaction	
Action items often is poorly disseminated and usually not followed through		Action items communicated and followed through thoroughly	X
Team members require frequent clarifications by PM for assigned tasks		Team members rarely require clarifications by PM for assigned tasks	X

Program Name B

Date Nov99

Pair choice section THREE: (People Management) choose most applicable term of the two for each row (page 2 of 2):

Leadership:

Long range organizational vision	X	Short term program and immediate work focus	
Lead through personal attention to others		Action-oriented leadership approach	X
Run as much of the organization as possible		Let team make decisions as much as possible	X
Direct and domineering style		Encourage independence in others	X
Traditional leaders respect hierarchy	X	Do what needs to be done	
Win cooperation rather than demand it	X	Tough-minded with others	
Act strongly and forcefully in the field of ideas	X	Prefer to lead other independent types while seeking autonomy for self	
Consults with team members to find solutions to problems	X	Consults team members to get validation of PM's predetermined solutions	
Keep people well informed	X	Only as much knowledge as necessary for their work	
Make things happen by focusing on the immediate problems		Long range focus and de-emphasize current problems	X
Manage others loosely and prefer minimal supervision	X	Follow traditional procedures and rules conscientiously	
Leadership, management decisions exclusively by program management		Program management makes decisions but gets inputs from team	X
Team-program manager relationship adult-adult	X	Team-program management relationship parent-child	
Program management makes decisions but gets inputs from team	X	All program team members responsible for program decisions	
When a problem arises: management takes over to solve it		Management lets the team solve the problems	X
Leadership is do as I say, not do as I do		Leadership by example	X
Program expectation not influenced by PM		Program expectation managed by PM	X
PM gives freedom to team, but does has no mentoring for members (abdication)		PM empowers teams by mentoring members to be leaders	X
Program management waits and sees what happens then plans		Management plans far in advance	X
Program management is constantly reacting to emergencies		Management is one step ahead of problems	X
Slow to decide on action/ indecisive at times		Take charge readily and often when required	X
Program management is complex, takes much time to understand	X	Management is simple, easy to figure out	
Program management prefers to plunge right in		Takes time to separate things to be done and order of doing them	X
Program management reacts spur of the moment		Methodically follows plans	X

Technical Competency of the program manager:

PM has technical experience particular to the particular s/w program		PM relies on team members solely	X
PM participates in technical reviews	X	PM only in non-technical reviews	
PM participates in making technical decisions when problems arise		PM delegates technical questions	X
PM does not get involved discussing technical options		PM contributes to technical options being discussed	X
PM does not review technical options and decisions		PM reviews technical options and decisions	X
PM actively attempts to keep up-to-date with current technology and standards	X	PM is removed from cutting edge technology issues	
PM receives technical periodicals and occasionally references applicable articles	X	PM doesn't read periodicals nor references current articles to team	
PM doesn't have technical background (or education)		PM has technical background (or education)	X
Team members avoid PM when they need technical advice		Team members generally consider talking to PM regarding technical issues	X

HR 13 **+ Comm.** 18 **+ Leadership** 20 **+ Tech. Competency** 7 **= People Mgmt. score** 58 **Enter on blk c.**

Program Name B

Date Nov99

Pair choice section Four: (Risk Management (RM)) choose most applicable term of the two for each row (page 1 of 2):

RM is formal and documented		RM is informal, if at all	X
A risk management plan exists	X	No risk management plan is developed	
RM is more of a data call than a useful document	X	RM drives decisions on the program	
RM is done prior to the program beginning		RM is done prior and during program execution	X
RM is only done during the program execution		RM is done prior and during program execution	X
Risks are generalized through the whole program		Risks are categorized	X
Risk management is done internally, only		An outside organization also contributes to the RM process	X
Risk is a management function		Risk is a program team function	X
Risk are precisely articulated		Risks are generalized, if at all	X
Each risk has a consequence		Consequences are generalized, if at all	X
A mitigation strategy is completed for each risk		Mitigation strategy is generalized, if at all	X
Contingency plans are developed for a RM plan		Contingency plans are ad hoc as problems arise in the program	X
Risks are anticipated		If problems arise, management will deal with it	X
The program doesn't have any risk		Programs that do not have risk, have problems	X
Risk management is automated		Risk management may use tools, but depend on human input	X
Risks are assigned probabilities	X	Probabilities are not relevant for RM	
All risks are potential problems, relative priorities for risks are not useful		Risks are weighed relative to other program risks and thus prioritized	X
Risk management information is only shared internally		Risk management information is shared with all stakeholders	X
Risk analysis uses ordinal rankings	X	Risk analysis uses actual measurements with a mathematical model	
Regret analysis used		No regret analysis done	X
Attach probabilities to future events		No probabilities associated with future events	X
Assessing risks with mechanical methods (simple spreadsheets, etc.)		Risks compared to other risks and sorted	X
Risk status tracked	X	Not tracked	
Technical risks examined	X	No technical risks examined	
Process risks examined	X	No process risks examined	
Product risks examined	X	No product risks examined	
Stakeholder/user risks examined	X	No examination of stakeholder/user risks	
Checklists used to identify risks	X	No checklists used	
Risks are tracked	X	No tracking or monitoring of risks	
Each risk has an impact		No impact analysis of risk	X
Each risk has a mitigation plan		No individual risk mitigation	X
Risks monitored by priority	X	No special attention to track higher priority risks	
Risk assessment is formalized		No formal risk assessment	X
Risk control is formalized		No formal risk control	X
Integration risks not considered		Integration risks examined	X

Risk Management page 1 of 2 score 21

Program Name B

Date Nov99

Pair choice section Four: (Risk Management (RM)) choose most applicable term of the two for each row (page 2 of 2):

Risks to cost	X	No cost risks examined	
Unforeseen risks have occurred in program		Any risk that came up had been identified previously	X
Personnel risks examined	X	No personnel risks examined	
Estimation risks examined	X	No estimation risks examined	
Planning risks examined	X	No planning risks examined	
Requirements risks examined	X	No requirements risks examined	
Resource risks examined	X	No resource risks examined	
Risk management plan updated regularly		No regular risk management plan updates	X
Risks charted		Risks not charted	X
Performance risks examined	X	Performance risks not examined	
Program management self risks examined		No program management risks examined	X
Risk from program constraints examined	X	No program constraint risks examined	
Each category of risks are prioritized	X	No prioritization	
Each category of risks are evaluated for impact	X	No impact analysis performed	
Each category of risks have control strategy		No control strategy	X
Documentation risks examined	X	No documentation risks examined	
Regret matrix tracked		No regret matrix or not tracked	X
Communication of risk activities are facilitated	X	No facilitation or promotion of communication of risk activities	
Taxonomy-based questionnaire used to identify risks		Taxonomy-based questionnaire not used	X
Associated hardware risks examined	X	No consideration for hardware risks	
Integration risks examined	X	Integration risks not examined	
Communication risks examined	X	Communication risks not examined	
Leadership risks examined	X	Leadership risks not considered	
Risk avoidance considered for certain risks	X	Risk avoidance not considered for risks	
Risk documentation forms used		No risk documentation forms used	X
Dependency risks examined	X	No dependency risks examined	
Alternatives like risk avoidance considered for high risk items	X	No consideration of risk avoidance	
Documented risk statements use a condition-consequence type format		Condition-consequence of risk statements not clearly defined	X
No assignment of ownership of risk mitigation action		Each risk mitigation action is assigned to an individual for resolution	X
Calculation of risk exposure made (probability X loss, for each risk)		No risk exposure calculations	X
Oral communication of risks only		Risks written in a way that communicates nature and status of factors	X
Triggers used to quantify risk conditions present		Risk conditions present are all subjective	X
Risk "czar" in program for monitoring risks		No special positions/responsibilities for risk monitoring	X
Post-program review completed (scheduled) for unanticipated problems ID		No post-program reviews completed or scheduled	X
No schedule risks examined		Risks to schedule investigated	X

Risk Management pg 2 of 2 score 23 **+ pg 1 of 2 score** 21 **= TOTAL SCORE** 44 **Enter on QMM scoresheet blk d.**

Program Name B

Date Nov99

Program Name: B

YES-NO-N/A Questionnaire

Date: Nov 99

No.	Requirements Management Questionnaire	Yes	No	N/A	
1	PM chose to have a formal requirements list	X			
2	Requirements recorded in some way	X			
3	Written requirements were part of some formal document	X			
4	Written requirements were informal		X		
5	At least some requirements were oral only		X		
6	All stakeholders were identified	X			
7	All stakeholders participated in the requirements extraction		X		
8	Some stakeholders participated in the requirements extraction	X			
9	Management extracted requirements, no stakeholder involvement		X		
10	Management passed requirements to development team	X			
11	Stakeholders not involved in Management extraction, but approves	X			
12	Management gets inputs from stakeholders, then develops requirments	X			
13	Developers work informally with users to arrive at requirements		X		
14	Same as 13, but management oversees and formalizes	X			
	If a waterfall or sequential development strategy:				
15	All requirements complete before design				
16	Some requirements left incomplete prior to design				
17	Requirements informal prior to design effort				
18	Requirements serve as input				
19	Length of time for requirements work greater than development work				
20	Requirements developed in parallel to design				
OR	***If a prototype, throwaway, or other development strategy:***				
15	Learn about requirements through development efforts	X			
16	No coding until all requirements are defined		X		
17	Requirements formal prior to design effort		X		
18	Requirements serve as output	X			
19	Requirements definition work in parallel to development efforts	X			
20	Requirements developed in parallel to design	X			
21	Are requirements frozen at some phase		X		
22	Change management exists	X			
23	Change management is formal	X			
24	Project strategy is consistent throughout development		X		
25	Requirements are updated	X			
26	Configuration Management (CM) exists	X			
27	CM is formal	X			
28	Requirements are testable	X			
29	Requirements testing considered/implemented during extraction	X			
30	Requirements testing plan exists	X			
31	Requirements testing is formal	X			
32	All requirements have priorities		X		
33	All requirements must be implemented	X			
34	Requirements are tested	X			
35	All requirements are equally important		X		
36	At least some requirements have priorities	X			
37	All requirements are traceable	X			
38	Traceability not important		X		
39	Each requirement has an author		X		
40	Who authored requirement is not important	X			
41	Initial set of requirements to be implemented, no requirements creep	X			
42	Structured and tracked changes to requirements only		X		
43	Change is inevitable, changes allowed at all times		X		
44	Change is inevitable, but changes limited	X			
45	Requirements control funding		X		
46	Requirements history kept	X			
47	Baseline established for requirements at some point prior to develop		X		
	TOTAL SCORING	36	3	0	39

Enter total score on QMM score sheet block e.

Program Name: B

YES-NO-N/A Questionnaire

Date: Nov 99

No.	Estimation/Planning Questionnaire	Yes	No	N/A	
1	A volume product metric used (LOC, # of files, # of screens, pages of doc)	X			
2	Measure used for various product elements (modules, components, CSCI)	X			
3	Product measures made by phase (amt at implementation, LOC changed at unit test)		X		
4	Other product attributes measured (FP, throughput, mem cap, cyclomatic complexity)	X			
5	Product metrics tracked and updated throughout program execution	X			
6	Event count process metric used (# defects in test, reqmt changes, milestones met)		X		
7	Time measure process metric used (cycle time)		X		
8	Process metrics tracked and updated throughout program execution		X		
9	Program cost estimations made from product or process metrics	X			
10	Program cost estimations tracked and updated to reflect progress/changes	X			
11	Factor analysis performed on program		X		
12	Program's primary purpose, including major functions and deliverables known	X			
13	Work breakdown structure developed	X			
14	Same as 13, but management oversees and formalizes		X		
15	Schedules developed based on realistic expectations	X			
16	Schedules tracked and updated based on new information	X			
17	Detailed activity lists used for clearly defined completed/not completed tasks		X		
18	Quality assurance plan or similar to aid in detecting defects early in program		X		
19	COCOMO estimates performed		X		
20	CSCI defined and tasked	X			
21	Estimates completed ad hoc		X		
22	Gantt charts used and updated	X			
23	Resource estimations (working hrs, job categories, task activities) done	X			
24	Earned value established	X			
25	Earned value tracked throughout program	X			
26	Quality expectations established for product with users and stakeholders	X			
27	Critical path for program tasks developed and tracked	X			
28	Meaure of effectiveness (MOE) or Figure of merit established and tracked	X			
29	Estimates are updated routinely	X			
30	Schedules are updated routinely	X			
31	Estimations are made by program management (top-down)	X			
32	Estimations are made by program team members (bottom-up)	X			
33	Automated program tracking used		X		
34	PM usually thorough in tracking and reporting schedules and financials	X			
35	WBS developed only as data call, not used in planning		X		
36	Earned value used to track program progress	X			
37	PM insists on prioritizing work reduction as schedule/funding compromised by stakeholders	X			
38	Estimations are done using both top down and bottoms up approaches	X			
39	All program team members involved in planning process	X			
40	Hardware also considered in estimation process	X			
41	Program history compiled		X		
42	System upgrades (SCR) software changes requests estimated individually		X		
43	Management duties apart of each team member's responsibilities	X			
44	PM dictates schedules to program team		X		
45	Code reviews planned in schedule	X			
46	Defined tangible milestones established for program tasks	X			
47	Test planning done at the start of the program		X		
48	Estimations are completed by those performing the tasks	X			
49	Sensitivity analysis performed for program choices		X		
50	Software deployment planning completed prior to development work		X		
	TOTAL SCORING	44	-8	0	36

Enter total score on QMM score sheet block f.

Program Name: B **YES-NO-N/A Questionnaire** Date: Nov 99

No.	People Management Questionnaire	Yes	No	N/A	
1	PM is accessable in person by each team member	X			
2	PM is accessable via email by each team member	X			
3	PM is accessable via phone by each team member	X			
4	PM not only considers a person's suitability, not also desire to be on a team	X			
5	PM consults with each team member regarding their career goals		X		
6	PM regularly holds meetings to inform team of program progress		X		
7	PM solicits opinions from team members before making decisions	X			
8	PM lets teams make decisions affecting their work	X			
9	PM frequently makes decisions without any consultation with members		X		
10	PM understands the technology/language of the program	X			
11	PM is able to communicate with other the technical issues in the program	X			
12	PM prioritizes problems or conflicts within the program	X			
13	PM assists team members in developing/advising of career path	X			
14	Same as 13, but management oversees and formalizes	X			
15	PM empowers program members to recommend firings of other members	X			
16	PM specifically assigns work to each program member		X		
17	PM sets communication protocol, which must be followed	X			
18	PM allows unrestricted communications	X			
19	PM readily makes tough decisions	X			
20	PM takes control in difficult/ problem areas	X			
21	PM looks ahead to new programs, new upgrades of existing program	X			
22	PM maintains regular communications with all stakeholders	X			
23	PM maintains regular communications with users	X			
24	PM encourages program team communication with users	X			
25	PM encourages program team communication with stakeholders	X			
26	PM facilitates horizontal communication within program	X			
27	PM facilitates communication during integration	X			
28	PM holds meetings without objectives listed prior to meeting		X		
29	PM must approve all decisions within the program		X		
30	PM must approve all interactions with stakeholders		X		
31	PM must approve all interactions with users		X		
32	PM makes all presentations to stakeholders/users		X		
33	PM is considered "flexible" in terms of program members personal issues	X			
34	PM, at least occasionally, schedules/promotes outside work team activities	X			
35	PM is readily willing to listen to program problems and complaints	X			
36	PM takes action to resolve program problems and complaints	X			
37	PM is generally respected by stakeholders, users, and organization	X			
38	PM sometimes fails to grasp important technical issues in program	X			
39	PM recruits program team members from outside organization	X			
40	PM directs what needs to be done and directs how to do it		X		
41	Program personnel have clearly defined specific tasks	X			
42	Although individual's tasks are specific, each exposed to the "bigger picture"	X			
43	PM has clearly defined his/her expectations for each individual	X			
44	PM delegation of duties is usually seemless in execution	X			
45	PM acts as facilitator to solving personnel conflicts	X			
46	PM attempts to motivate individuals on the program team	X			
47	PM clearly separates technical from managerial roles for individuals	X			
48	PM directs how he/she expects the task to be accomplished		X		
49	PM directs what needs to be done, but does not direct how	X			
50	PM attempts to spotlight individuals in the program for positive exposure	X			
	TOTAL SCORING	42	8	0	50

Enter total score on QMM score sheet block g.

Program Name: B

YES-NO-N/A Questionnaire

Date: Nov 99

No.	Risk Management Questionnaire	Yes	No	N/A	
1	Risk Management (RM) is specifically an activity in the program	X			
2	RM is formal and documented	X			
3	A specific RM plan exists		X		
4	RM is required in the program, but not used during the program		X		
5	RM is done prior to the program execution	X			
6	RM is done by an outside entity to the development	X			
7	RM is done internally only		X		
8	RM is both internally performed and externally assessed		X		
9	RM planning occurs during or after major milestones in the program	X			
10	Risk Assessment is only a management function		X		
11	RM is informal or non existent		X		
12	There is a RM plan, but it is not updated or tracked	X			
13	Risks are only generalized	X			
14	Same as 13, but management oversees and formalizes	X			
15	Each risk has a consequence	X			
16	Each risk has a likelihood rating of some sort	X			
17	Each risk has a mitigation strategy		X		
18	Risk Management is automated		X		
19	Risks are tracked		X		
21	Regret analysis performed	X			
22	RM drives decisions in the program	X			
23	Risks have probabilities	X			
24	Risk Management is ad hoc		X		
25	RM information is shared with all stakeholders (as appropriate)	X			
26	Risks are weighed relative to other program risks	X			
27	Risk Assessment is a program team activity	X			
28	Risk Assessment done prior to program start	X			
29	Risk Assessment includes personnel risk	X			
30	RM uses tools, but depends on human decisions	X			
31	Risk Assessment includes cost risks	X			
32	Risk Assessment includes schedule risks	X			
33	Risk Assessment includes technology risks	X			
34	Risk Assessment is briefed organization structure above program manager	X			
35	Risk Assessment includes requirements risks	X			
36	Risk Assessment includes user risks (too little involvement of user)	X			
37	Risk Assessment includes documentation risks	X			
38	Risk Assessment includes integration risks	X			
39	Risk Assessment includes interface risks (non-standard)	X			
40	Risk Assessment includes continuing requirements change (feature creep)	X			
41	Risk Assessment includes dependent projects/programs risks	X			
42	Documentation proof exists to demonstrate following risk management plan		X		
43	High risk have measured tracking (high profile status)		X		
44	Organizational history used to search for risks		X		
45	Other organizational checklists used for risk assessment		X		
46	Internal organizational checklists used for risk assessment		X		
47	Risk Assessment information contributed to internal or other database		X		
48	Risk Assessment includes internal organization risks	X			
49	Risk Assessment includes stakeholder risks	X			
50	No risk management needed; program is straightforwarded & understood		X		
	TOTAL SCORING	41	2	0	43

Enter total score on QMM score sheet block h.

Pair choice section ONE: (Requirements Management) choose most applicable term of the two for each row (page 1 of 2):

formal requirement list		informal requirement list	X
written requirements		oral requirements	X
requirements informal, but recorded	X	requirements not recorded	
requirements as part of an SRS (or other formal repository)		requirements informally recorded	X
requirements taken as is from customer	X	look to reformulate, interview in-depth, or otherwise re-validate	
only one development strategy used		strategies not consistent, used at different times	X
stakeholders as part of requirements development	X	stakeholders approving requirements after formulated by development team	
requirements are testable		requirements have no test plans	X
informal test plan or no test plan	X	formal test plan	
test team involved with requirements		no test team input or plans during requirements development	X
only a percentage of requirements present in baseline	X	baseline must contain all requirements	
requirements documentation has hierarchical structure		all requirements must be implemented	X
requirements have listed responsible party		requirements origin not important	X
requirements documentation have versions		no requirements history	X
requirements have specific attribute values		requirements all rank evenly	X
funding controls requirements definition	X	requirements definition controls funding	
requirements are top down	X	requirements are bottom up	
users/stakeholders are identified and interviewed (market survey)		no special consideration to identify users/stakeholders	X
each requirement has a singular concept		some requirements are compound statements	X
requirements definition minimized when funding short	X	program scope may reduce, but requirements definition completed	
requirements extraction has formal process		requirements extraction ad hoc	X
change procedures formal		change procedures ad hoc	X
users/stakeholders somehow involved in requirements definition		program team only involved in requirements definition	X
management sets requirements for developers	X	developers at least partially involved in setting requirements	
requirements changed at least once since baseline established prior to new version	X	requirements in baseline has not changed prior to new version or upgrade	
no ranking of requirements	X	requirements have priorities assigned	
use-case diagrams (or other models or scenario developments)		no models used for requirements extraction	X
requirements changes informal	X	requirements changes formal	
plan to "freeze" requirements as some designated milestone		no provision for "freezing" requirements	X
requirements must be traceable		origin of requirements not important	X
requirements must be testable		system developed must be testable	X
test plans to determine requirements implemented		no test plans needed for requirements verification	X
requirements have priorities in implementation		all requirements must be implemented	X
some requirements have multiple statements or ideas	X	one idea, one statement per requirement	

Requirements Management (page 1 of 2) score 7

Program Name C

Date Nov 99

ANSWER THIS BLOCK OF QUESTIONS ONLY IF A SEQUENTIAL OR WATERFALL APPROACH IS USED FOR DEVELOPMENT (Requirements page 2 of 2)			
requirements first, then initial development work		initial development work then requirements	
requirements documentation driving development		requirements documentation developed in parallel/after development	
user feedback considered during development		after development starts, user feedback serves as input to new work	
change management procedures used strictly		change management procedures as guidance only	
design decisions prior to or in parallel to requirements development		design decisions only after approved requirements stabilized	
requirements summarized what we have developed		requirements are the blueprint for development	
length of time for requirements work greater than development work		length of time for requirements work less than development work	
requirements have design detail		no design detail in requirements	
requirements creep to be avoided		requirements creep o.k., but need to be controlled	
freeze requirements at some point		requirements are fluid throughout development	
formal change procedure		informal change procedure	
change management plan		no change management plan	
requirements ambiguity always present to some extent		requirements ambiguity unacceptable at any level	
testing considered up front during requirements determination		testing considered down the line during development	
requirements development team members different from implementation		those working on requirements, work on implementation	
start implementation as early as possible to help define requirements		requirements must be defined prior to any implementation work	
ANSWER THIS BLOCK OF QUESTIONS ONLY IF A PROTOTYPING, THROWAWAY, SYNCHRONIZE & STABILIZE, OR OTHER STRATEGY USED			
develop prototype, then determine requirements	X	determine requirements prior to any development work	
requirements testing done after each iteration	X	no testing	
individual changes as necessary	X	only block changes made	
development team decides on changes after iteration	X	users involved with changes	
changes based on feedback only from user for correction of problems	X	changes to upgrade system and correct problems	
funding controls changes and change procedures	X	changes control funding	
requirements documentation finalized prior to development		requirements fluid throughout development (only freeze at end)	X
requirements test plans completed prior to development		requirements test plans completed after development	X
requirements first, then initial development work		initial development work then requirements	X
use development effort to learn more about requirements	X	define all requirements prior to coding anything	
requirements ambiguity always present to some extent	X	requirements ambiguity unacceptable at any level	
requirements have design detail	X	no design detail in requirements	
user feedback considered during development	X	after development starts, user feedback serves as input to new work	
get something to users as soon as possible for evaluation	X	make sure it is complete before releasing	
management dictates requirements	X	developm't team visually represent requirements through rapid prototyping	
new requirements allowed after initial requirements defined	X	new requirements not allowed	

Requirements Management (pg 2 of 2) score 16 **+ pg 1 score** 7 **= TOTAL SCORE** 23 **Enter on QMM scoresheet blk a.**

Program Name C

Date Nov 99

Pair choice section TWO: (Estimation/Planning Management) choose most applicable term of the two for each row (page 1 of 2):

At least one estimation method used in program		No estimates	X
Formal derivation of product metric for estimation of size		Ad hoc size estimation	X
Ad hoc process evaluation	X	Formal derivation of at least one process metric	
Develop work breakdown structure (WBS)		Assign work as needs arise	X
Estimates are developed to fulfill a data call only	X	Use estimates to plan program	
Use estimates to sell program only	X	Estimates are useful to the project team for planning purposes	
Resource evaluations made for program	X	No resource evaluation for planning	
Use both bottom up & top down for estimate, use one stakeholders like	X	Use both bottom up & top down and evaluate significant differences	
Estimates made and not updated	X	Estimates updated throughout program	
Resource estimations used to adjust product size estimate	X	Estimations made irregardless of resources available	
Estimations made to fit budget	X	Budget made from estimations	
Estimations compromised to get program	X	Rather risk loss of program than compromise confident estimations	
Cycle time estimations		No cycle time estimations	X
Event count estimations		No event count estimations	X
Lines of code (LOC) estimation		No LOC estimation	X
Function Point (FP) estimation		No FP estimation	X
Estimates by algorithmic methods		Estimates by analogy	X
Expert judgement for estimation		Ad hoc estimates	X
Estimates by algorithmic methods		Ad hoc estimates	X
Expert judgement for estimates		Estimates by analogy	X
Ad hoc estimates	X	Estimates by analogy	
Bottom up estimates	X	Expert judgement	
Top down estimates	X	Expert judgement	
Ad hoc estimates	X	Any other estimate process	
Fuzzy logic estimating method		No formal estimation methodology	X
WBS development from estimates		WBS development in parallel or prior to estimation completion	X
Critical path of program determined		Tasks developed but no path is identified	X
Estimators are program team members	X	Estimators are outside program team	
Management only on estimations	X	All team members involved in estimation process	
Estimates updated at reviews		No updates of estimates	X
Estimates updated at reviews	X	Estimates constantly updates (in between reviews, too)	
Estimate procedures stay the same		Estimate procedures change	X
Stakeholders are part of estimation process		Stakeholders briefed on estimations after completion	X
Estimates are used beyond initial selling of program		Estimates are one time events, used for a specific purpose once	X
WBS has objective measure of completeness		Important to have WBS as guide, not rigid implementation	X

Estimation/Planning Management page 1 of 2 score 7

Program Name__C__

Date Nov 99

Pair choice section TWO: (Estimation/Planning Management) choose most applicable term of the two for each row (page 2 of 2):

Life cycle estimates		Estimates for program initiation only	X
System upgrades (SCR) software change requests estimated individually		Systems upgrades estimated as whole	X
Estimates for on-going resources needed to maintain s/w		Estimates for maintenance not done	X
Informal re-estimates during development	X	Formal re-estimates at pre-defined milestones	
Formal re-estimates when amendment changing the system is introduced		Informal re-estimates when amendment changing the system	X
Person in-charge of estimation walks in a managers office to get an opinion	X	Meeting(s) organized for purpose of performing cost estimations	
Factor analysis prior to commencement of program		None done	X
Change control procedures set in place		No set procedures	X
Elapsed time and actual work time estimates	X	One or the other or neither	
No schedule created	X	Schedule created	
Schedule not updated		Schedule updated	X
Schedule followed	X	Schedule not followed	
Tasks identification arises as program progresses	X	Detailed level tasks identified prior to program initiation	
Scope of program understood by all		Scope not explicitly defined	X
Quality factors and criteria identified		No explicit quality factors defined	X
No project tracking tools used		Project tracking tools used	X
CSCIs identified and tasked		CSCIs not explicitly identified	X
Expectations are managed via estimations		Estimations are made to fit preconceived expectations	X
No cost schedule developed	X	Cost schedule developed	
No resource schedule developed	X	Resource schedule developed	
Team members, management know at any time if in budget & schedule		Exact budget & schedule status somewhat unclear to at least some	X
Individual program phases are estimated		Only top level program estimated	X
Stakeholders/users emphasis understood- quick to field or all complete		Program management sets delivery tradeoffs without outside input	X
Testing planned with initial program planning		Testing not in initial planning	X
Documentation not considered in initial planning	X	Documentation part of initial planning	
Hardware considered in estimations		Software only considered	X
No formal schedule/cost tracking	X	Formal procedures established for tracking cost and schedule	
Earned value set up		Earned value not used	X
Estimations omit documentation planning	X	Documentation in estimates	
Training omitted in estimates		Training part of estimates	X
Earned value set up, but not tracked	X	Earned value set up and tracked	
Detailed planning done with incomplete set of requirements	X	Detailed planning done with detailed set of requirements	
Complete infrastructure support mechanism understood for estimations		No consideration of infrastructure done for estimations	X
Team possibilities considered for planning of program		No consideration for outside teaming possibilities	X
Work Breakdown Structure (WBS) set up		No WBS completed	X

Estimation/Planning Management pg 2 of 2 score 4 **+ pg 1 of 2 score** 7 **= TOTAL SCORE** 11 **Enter QMM scoresheet blk b.**

Program Name C

Date Nov 99

Pair choice section THREE: (People Management) choose most applicable term of the two for each row (page 1 of 2):

Human Resources

Program team members have clearly defined, segmented roles		Work responsibilities are shared	X
Formal team building procedures are used		No formal team building emphasized	X
Program manager flexible regarding work hours	X	Program manager maintains strict standards for work hours	
Big picture conveyed to all team members by program management	X	Program management focuses on the partitioned tasks with team	
People issues dealt with primarily through indirect methods (email, memo, etc)		People issues dealt with primarily through direct methods (face-to-face)	X
Training is required and planned on a regular basis		Training is ad hoc	X
Each team member is educated on and understands overall program and their roles		Team members only know their respective areas	X
Consideration for team members' career goals are reflected in assignments		Team members must adapt to tasks that are assigned	X
Team members assignments and responsibilities are mostly dictated by PM	X	Assignments and responsibilities are discussed and agreed upon with PM	
Management leads in problem solving	X	Management facilitates and lets team lead in problem solving	
Management welcomes problems as challenges and opportunities	X	Management views problems as obstacles and grounds for punishment	
Team members participate in performance evaluations of peers		Personnel evaluations are strictly PM responsibility	X
Management reinforcement feedback sparse and inconsistent, if any		Management provides timely reinforcement feedback for positive behaviors	X
Management provides basic needs of office facilities fairly well	X	Office facilities are a drawback to working in the program	
Working conditions are fairly comfortable, time off policy "flexible"	X	Working conditions and time off policy is inconsistent and difficult at times	

Communication:

Communications primarily written (email, memo, etc.)		Communications primarily verbal (face-to-face)	X
Detailed instructions: oral presentation, follow-up email	X	Email or memo only	
Formal communication protocol		Informal communications	X
External vertical communications restricted		External vertical communication allowed	X
Coders notebook, weekly accomplishment reports required		Not required	X
User-coder relationship established, encouraged, and mediated	X	User-coder interaction minimized	
Meetings structured to minimize wasted time	X	Meetings unstructured and open ended	
Meetings have agenda, objectives, and conclude with action items	X	Meeting agenda fluid and open ended	
Program management and coder communication face to face	X	Program management and coder communication primarily email	
Program team updated regularly regarding organizational & program status	X	Meetings infrequently scheduled	
Open communications is encouraged	X	Communication through chain of command only is encouraged	
Program manager accessible for discussions	X	Program manager difficult to get an appointment to see	
Program management (PM) is viewed as separate from team		PM mixes with team frequently	X
Management regularly holds team meetings		Meetings are sporadic	X
Meetings are structured with definite goals and objectives	X	Meetings are informal	
Program management is generally easy to reach and talk to	X	PM is usually hard to get a hold of and difficult to talk to	
Team-program manager relationship adult-adult	X	Team-program management relationship parent-child	
Schedules are spontaneous and poorly communicated	X	Schedules must be fixed and rigidly followed and formally reported	
Work is seen as complex processes involving team working together	X	Work broken into pieces with minimal team member interaction	
Action items sometimes are not followed through	X	Action items communicated and followed through thoroughly	
Team members require frequent clarifications by PM for assigned tasks	X	Team members rarely require clarifications by PM for assigned tasks	

Program Name__C__

Date Nov 99

Pair choice section THREE: (People Management) choose most applicable term of the two for each row (page 2 of 2):

Leadership:

Long range organizational vision		Short term program and immediate work focus	X
Lead through personal attention to others		Action-oriented leadership approach	X
Run as much of the organization as possible		Let team make decisions as much as possible	X
Direct and domineering style		Encourage independence in others	X
Traditional leaders respect hierarchy		Do what needs to be done	X
Win cooperation rather than demand it	X	Tough-minded with others	
Act strongly and forcefully in the field of ideas	X	Prefer to lead other independent types while seeking autonomy for self	
Consults with team members to find solutions to problems	X	Consults team members to get validation of PM's predetermined solutions	
Keep people well informed	X	Only as much knowledge as necessary for their work	
Make things happen by focusing on the immediate problems	X	Long range focus and de-emphasize current problems	
Manage others loosely and prefer minimal supervision	X	Follow traditional procedures and rules conscientiously	
Leadership, management decisions exclusively by program management		Program management makes decisions but gets inputs from team	X
Team-program manager relationship adult-adult	X	Team-program management relationship parent-child	
Program management makes decisions but gets inputs from team	X	All program team members responsible for program decisions	
When a problem arises: management takes over to solve it		Management lets the team solve the problems	X
Leadership is do as I say, not do as I do		Leadership by example	X
Program expectation not influenced by PM		Program expectation managed by PM	X
PM gives freedom to team, but does has no mentoring for members	X	PM empowers teams by mentoring members to be leaders	
Program management waits and sees what happens then plans		Management plans far in advance	X
Program management reacts to emergencies		Management is one step ahead of problems	X
Slow to decide on action		Take charge readily and often when required	X
Program management is complex, takes much time to understand	X	Management is simple, easy to figure out	
Program management prefers to plunge right in		Takes time to separate things to be done and order of doing them	X
Program management reacts to needs of the moment		Methodically follows plans	X

Technical Competency of the program manager:

PM has technical experience particular to the particular s/w program	X	PM relies on team members solely	
PM participates in technical reviews	X	PM only in non-technical reviews	
PM participates in making technical decisions when problems arise	X	PM delegates technical questions	
PM does not get involved discussing technical options		PM contributes to technical options being discussed	X
PM does not review technical options and decisions		PM reviews technical options and decisions	X
PM actively attempts to keep up-to-date with current technology and standards	X	PM is removed from cutting edge technology issues	
PM receives technical periodicals and occasionally references applicable articles	X	PM doesn't read periodicals nor references current articles to team	
PM doesn't have technical background (or education)		PM has technical background (or education)	X
Team members avoid PM when they need technical advice		Team members generally consider talking to PM regarding technical issues	X

HR 8 **+ Comm.** 16 **+ Leadership** 19 **+ Tech. Competency** 9 **= People Mgmt. score** 52 **Enter on blk c.**

Program Name__C__

Date Nov 99

Pair choice section Four: (Risk Management (RM)) choose most applicable term of the two for each row (page 1 of 2):

RM is formal and documented		RM is informal, if at all	X
A risk management plan exists		No risk management plan is developed	X
RM is more of a data call than a useful document	X	RM drives decisions on the program	
RM is done prior to the program beginning		RM is done prior and during program execution	X
RM is only done during the program execution	X	RM is done prior and during program execution	
Risks are generalized through the whole program	X	Risks are categorized	
Risk management is done internally, only	X	An outside organization also contributes to the RM process	
Risk is a management function	X	Risk is a program team function	
Risk are precisely articulated		Risks are generalized, if at all	X
Each risk has a consequence		Consequences are generalized, if at all	X
a mitigation strategy is completed for each risk		Mitigation strategy is generalized, if at all	X
Contingency plans are developed for a RM plan		Contingency plans are ad hoc as problems arise in the program	X
Risks are anticipated		If problems arise, management will deal with it	X
The program doesn't have any risk		Programs that do not have risk, have problems	X
Risk management is automated		Risk management may use tools, but depend on human input	X
Risks are assigned probabilities		Probabilities are not relevant for RM	X
All risks are potential problems, relative priorities for risks are not useful	X	Risks are weighed relative to other program risks and thus prioritized	
Risk management information is only shared internally	X	Risk management information is shared with all stakeholders	
Risk analysis uses ordinal rankings	X	Risk analysis uses actual measurements with a mathematical model	
Regret analysis used		No regret analysis done	X
Attach probabilities to future events		No probabilities associated with future events	X
Assessing risks with mechanical methods (simple spreadsheets, etc.)		Risks compared to other risks and sorted	X
Risk status tracked		Not tracked	X
Technical risks examined	X	No technical risks examined	
Process risks examined		No process risks examined	X
Product risks examined		No product risks examined	X
Stakeholder/user risks examined		No examination of stakeholder/user risks	X
Checklists used to identify risks		No checklists used	X
Risks are tracked		No tracking or monitoring of risks	X
Each risk has an impact		No impact analysis of risk	X
Each risk has a mitigation plan		No individual risk mitigation	X
Risks monitored by priority		No special attention to track higher priority risks	X
Risk assessment is formalized		No formal risk assessment	X
Risk control is formalized		No formal risk control	X
Integration risks not considered	X	Integration risks examined	

Risk Management page 1 of 2 score 6

Program Name C

Date Nov 99

Pair choice section Four: (Risk Management (RM)) choose most applicable term of the two for each row (page 2 of 2):

Risks to cost	X	No cost risks examined	
Unforeseen risks have occurred in program	X	Any risk that came up had been identified previously	
Personnel risks examined		No personnel risks examined	X
Estimation risks examined	X	No estimation risks examined	
Planning risks examined		No planning risks examined	X
Requirements risks examined		No requirements risks examined	X
Resource risks examined	X	No resource risks examined	
Risk management plan updated regularly		No regular risk management plan updates	X
Risks charted		Risks not charted	X
Performance risks examined		Performance risks not examined	X
Program management self risks examined		No program management risks examined	X
Risk from program constraints examined		No program constraint risks examined	X
Each category of risks are prioritized		No prioritization	X
Each category of risks are evaluated for impact		No impact analysis performed	X
Each category of risks have control strategy		No control strategy	X
Documentation risks examined		No documentation risks examined	X
Regret matrix tracked		No regret matrix or not tracked	X
Communication of risk activities are facilitated		No facilitation or promotion of communication of risk activities	X
Taxonomy-based questionnaire used to identify risks		Taxonomy-based questionnaire not used	X
Associated hardware risks examined	X	No consideration for hardware risks	
Integration risks examined		Integration risks not examined	X
Communication risks examined		Communication risks not examined	X
Leadership risks examined		Leadership risks not considered	X
Risk avoidance considered for certain risks	X	Risk avoidance not considered for risks	
Risk documentation forms used		No risk documentation forms used	X
Dependency risks examined		No dependency risks examined	X
Alternatives like risk avoidance considered for high risk items	X	No consideration of risk avoidance	
Documented risk statements use a condition-consequence type format		Condition-consequence of risk statements not clearly defined	X
No assignment of ownership of risk mitigation action	X	Each risk mitigation action is assigned to an individual for resolution	
Calculation of risk exposure made (probability X loss, for each risk)		No risk exposure calculations	X
Oral communication of risks only	X	Risks written in a way that communicates nature and status of factors	
Triggers used to quantify risk conditions present		Risk conditions present are all subjective	X
Risk "czar" in program for monitoring risks		No special positions/responsibilities for risk monitoring	X
Post-program review completed (scheduled) for unanticipated problems ID		No post-program reviews completed or scheduled	X
No schedule risks examined	X	Risks to schedule investigated	

Risk Management pg 2 of 2 score 6 **+ pg 1 of 2 score** 6 **= TOTAL SCORE** 12 **Enter on QMM scoresheet blk d.**

Program Name C

Date Nov 99

Program Name: C

YES-NO-N/A Questionnaire

Date: Nov 99

No.	Requirements Management Questionnaire	Yes	No	N/A	
1	PM chose to have a formal requirements list		x		
2	Requirements recorded in some way	x			
3	Written requirements were part of some formal document		x		
4	Written requirements were informal	x			
5	At least some requirements were oral only	x			
6	All stakeholders were identified		x		
7	All stakeholders participated in the requirements extraction		x		
8	Some stakeholders participated in the requirements extraction	x			
9	Management extracted requirements, no stakeholder involvement		x		
10	Management passed requirements to development team		x		
11	Stakeholders not involved in Management extraction, but approves		x		
12	Management gets inputs from stakeholders, then develops requirments		x		
13	Developers work informally with users to arrive at requirements	x			
14	Same as 13, but management oversees and formalizes		x		
	If a waterfall or sequential development strategy:				
15	All requirements complete before design				
16	Some requirements left incomplete prior to design				
17	Requirements informal prior to design effort				
18	Requirements serve as input				
19	Length of time for requirements work greater than development work				
20	Requirements developed in parallel to design				
OR	***If a prototype, throwaway, or other development strategy:***				
15	Learn about requirements through development efforts	x			
16	No coding until all requirements are defined		x		
17	Requirements formal prior to design effort		x		
18	Requirements serve as output	x			
19	Requirements definition work in parallel to development efforts	x			
20	Requirements developed in parallel to design		x		
21	Are requirements frozen at some phase		x		
22	Change management exists		x		
23	Change management is formal		x		
24	Project strategy is consistent throughout development		x		
25	Requirements are updated		x		
26	Configuration Management (CM) exists		x		
27	CM is formal		x		
28	Requirements are testable	x			
29	Requirements testing considered/implemented during extraction		x		
30	Requirements testing plan exists		x		
31	Requirements testing is formal		x		
32	All requirements have priorities		x		
33	All requirements must be implemented	x			
34	Requirements are tested	x			
35	All requirements are equally important	x			
36	At least some requirements have priorities	x			
37	All requirements are traceable		x		
38	Traceability not important		x		
39	Each requirement has an author		x		
40	Who authored requirement is not important	x			
41	Initial set of requirements to be implemented, no requirements creep		x		
42	Structured and tracked changes to requirements only		x		
43	Change is inevitable, changes allowed at all times	x			
44	Change is inevitable, but changes limited		x		
45	Requirements control funding		x		
46	Requirements history kept		x		
47	Baseline established for requirements at some point prior to develop		x		
	TOTAL SCORING	11	-10	0	1

Enter total score on QMM score sheet block e.

Program Name: C

YES-NO-N/A Questionnaire

Date: Nov 99

No.	Estimation/Planning Questionnaire	Yes	No	N/A	
1	A volume product metric used (LOC, # of files, # of screens, pages of doc)		x		
2	Measure used for various product elements (modules, components, CSCI)		x		
3	Product measures made by phase (amt at implementation, LOC changed at unit test)		x		
4	Other product attributes measured (FP, throughput, mem cap, cyclomatic complexity)		x		
5	Product metrics tracked and updated throughout program execution		x		
6	Event count process metric used (# defects in test, reqmt changes, milestones met)		x		
7	Time measure process metric used (cycle time)		x		
8	Process metrics tracked and updated throughout program execution		x		
9	Program cost estimations made from product or process metrics		x		
10	Program cost estimations tracked and updated to reflect progress/changes		x		
11	Factor analysis performed on program		x		
12	Program's primary purpose, including major functions and deliverables known		x		
13	Work breakdown structure developed		x		
14	Task estimated with realistic expectations of productivity probabilities	x			
15	Schedules developed based on realistic expectations	x			
16	Schedules tracked and updated based on new information		x		
17	Detailed activity lists used for clearly defined completed/not completed tasks		x		
18	Quality assurance plan or similar to aid in detecting defects early in program		x		
19	COCOMO estimates performed		x		
20	CSCI defined and tasked		x		
21	Estimates completed ad hoc	x			
22	Gantt charts used and updated		x		
23	Resource estimations (working hrs, job categories, task activities) done		x		
24	Earned value established		x		
25	Earned value tracked throughout program		x		
26	Quality expectations established for product with users and stakeholders	x			
27	Critical path for program tasks developed and tracked		x		
28	Meaure of effectiveness (MOE) or Figure of merit established and tracked		x		
29	Estimates are updated routinely		x		
30	Schedules are updated routinely		x		
31	Estimations are made by program management (top-down)	x			
32	Estimations are made by program team members (bottom-up)		x		
33	Automated program tracking used		x		
34	PM usually thorough in tracking and reporting schedules and financials	x			
35	WBS developed only as data call, not used in planning	x			
36	Earned value used to track program progress		x		
37	PM insists on prioritizing work reduction as schedule/funding compromised by stakeholders	x			
38	Estimations are done using both top down and bottoms up approaches	x			
39	All program team members involved in planning process		x		
40	Hardware also considered in estimation process		x		
41	Program history compiled		x		
42	System upgrades (SCR) software changes requests estimated individually		x		
43	Management duties apart of each team member's responsibilities	x			
44	PM dictates schedules to program team	x			
45	Code reviews planned in schedule		x		
46	Defined tangible milestones established for program tasks		x		
47	Test planning done at the start of the program		x		
48	Estimations are completed by those performing the tasks	x			
49	Sensitivity analysis performed for program choices		x		
50	Software deployment planning completed prior to development work		x		
	TOTAL SCORING	4	-24	0	-20

Enter total score on QMM score sheet block f.

Program Name: C

YES-NO-N/A Questionnaire

Date: Nov 99

No.	People Management Questionnaire	Yes	No	N/A	
1	PM is accessable in person by each team member	x			
2	PM is accessable via email by each team member	x			
3	PM is accessable via phone by each team member	x			
4	PM not only considers a person's suitability, not also desire to be on a team		x		
5	PM consults with each team member regarding their career goals	x			
6	PM regularly holds meetings to inform team of program progress	x			
7	PM solicits opinions from team members before making decisions	x			
8	PM lets teams make decisions affecting their work	x			
9	PM frequently makes decisions without any consultation with members		x		
10	PM understands the technology/language of the program	x			
11	PM is able to communicate with other the technical issues in the program	x			
12	PM prioritizes problems or conflicts within the program	x			
13	PM assists team members in developing/advising of career path	x			
14	PM empowers program members to recommend hiring of other members		x		
15	PM empowers program members to recommend firing of other members		x		
16	PM specifically assigns work to each program member	x			
17	PM sets communication protocol to be followed	x			
18	PM allows unrestricted communications	x			
19	PM readily makes tough decisions		x		
20	PM takes control in difficult/ problem areas	x			
21	PM looks ahead to new programs, new upgrades of existing program	x			
22	PM maintains regular communications with all stakeholders	x			
23	PM maintains regular communications with users	x			
24	PM encourages program team communication with users	x			
25	PM encourages program team communication with stakeholders	x			
26	PM facilitates horizontal communication within program	x			
27	PM facilitates communication during integration	x			
28	PM holds meetings without objectives listed prior to meeting		x		
29	PM must approve all decisions within the program	x			
30	PM must approve all interactions with stakeholders		x		
31	PM must approve all interactions with users		x		
32	PM makes all presentations to stakeholders/users		x		
33	PM is considered "flexible" in terms of program members personal issues		x		
34	PM, at least occasionally, schedules/promotes outside work team activities	x			
35	PM is readily willing to listen to program problems and complaints	x			
36	PM takes action to resolve program problems and complaints	x			
37	PM is generally respected by stakeholders, users, and organization	x			
38	PM sometimes fails to grasp important technical issues in program		x		
39	PM recruits program team members from outside organization	x			
40	PM directs what needs to be done and directs how to do it	x			
41	Program personnel have clearly defined specific tasks	x			
42	Although individual's tasks are specific, each exposed to the "bigger picture"	x			
43	PM has clearly defined his/her expectations for each individual	x			
44	PM delegation of duties is usually seemless in execution	x			
45	PM acts as facilitator to solving personnel conflicts	x			
46	PM attempts to motivate individuals on the program team	x			
47	PM clearly separates technical from managerial roles for individuals		x		
48	PM directs how he/she expects the task to be accomplished	x			
49	PM directs what needs to be done, but does not direct how	x			
50	PM attempts to spotlight individuals in the program for positive exposure	x			
	TOTAL SCORING	42	6	0	48

Enter total score on QMM score sheet block g.

Program Name: C

YES-NO-N/A Questionnaire

Date: Nov 99

No.	Risk Management Questionnaire	Yes	No	N/A	
1	Risk Management (RM) is specifically an activity in the program		x		
2	RM is formal and documented		x		
3	A specific RM plan exists		x		
4	RM is required in the program, but not used during the program		x		
5	RM is done prior to the program execution		x		
6	RM is done by an outside entity to the development		x		
7	RM is done internally only		x		
8	RM is both internally performed and externally assessed		x		
9	RM planning occurs during or after major milestones in the program		x		
10	Risk Assessment is only a management function	x			
11	RM is informal or non existent		x		
12	There is a RM plan, but it is not updated or tracked		x		
13	Risks are only generalized	x			
14	Each risk is delineated		x		
15	Each risk has a consequence		x		
16	Each risk has a likelihood rating of some sort		x		
17	Each risk has a mitigation strategy		x		
18	Risk Management is automated		x		
19	Risks are tracked		x		
21	Regret analysis performed	x			
22	RM drives decisions in the program		x		
23	Risks have probabilities		x		
24	Risk Management is ad hoc	x			
25	RM information is shared with all stakeholders (as appropriate)		x		
26	Risks are weighed relative to other program risks		x		
27	Risk Assessment is a program team activity		x		
28	Risk Assessment done prior to program start		x		
29	Risk Assessment includes personnel risk		x		
30	RM uses tools, but depends on human decisions		x		
31	Risk Assessment includes cost risks		x		
32	Risk Assessment includes schedule risks		x		
33	Risk Assessment includes technology risks		x		
34	Risk Assessment is briefed organization structure above program manager		x		
35	Risk Assessment includes requirements risks		x		
36	Risk Assessment includes user risks (too little involvement of user)		x		
37	Risk Assessment includes documentation risks		x		
38	Risk Assessment includes integration risks		x		
39	Risk Assessment includes interface risks (non-standard)		x		
40	Risk Assessment includes continuing requirements change (feature creep)		x		
41	Risk Assessment includes dependent projects/programs risks		x		
42	Documentation proof exists to demonstrate following risk management plan		x		
43	High risk have measured tracking (high profile status)		x		
44	Organizational history used to search for risks		x		
45	Other organizational checklists used for risk assessment		x		
46	Internal organizational checklists used for risk assessment		x		
47	Risk Assessment information contributed to internal or other database		x		
48	Risk Assessment includes internal organization risks		x		
49	Risk Assessment includes stakeholder risks		x		
50	No risk management needed; program is straightforwarded & understood		x		
	TOTAL SCORING	-2	-19	0	-21

Enter total score on QMM score sheet block h.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C
FINAL SURVEY FORMS TEMPLATE WITH SCORING

Pair choice section ONE: (Requirements Management) choose most applicable term of the two for each row (page 1 of 2):

Formal requirement list	2	Informal requirement list	1
Written requirements	2	Oral requirements	0
Requirements informal, but recorded	1	Requirements not recorded	0
Requirements as part of an SRS (or other formal repository)	2	Requirements informally recorded	1
Requirements taken as is from customer	0	Look to reformulate, interview in-depth, or otherwise re-validate	2
Only one development strategy used	1	Strategies not consistent, used at different times	0
Stakeholders as part of requirements development	2	Stakeholders approve requirements after formulated by development team	1
Requirements are testable	2	Requirements have no test plans	0
Informal test plan or no test plan	0	Formal test plan	2
Test team involved with requirements	1	No test team input or plans during requirements development	0
Only a percentage of requirements present in baseline	0	Baseline must contain all requirements	2
Requirements documentation has hierarchical structure	1	All requirements must be implemented	0
Requirements have listed responsible party	1	Requirements origin not important	0
Requirements documentation have versions	2	No requirements history	0
Requirements have specific attribute values	1	Requirements all rank evenly	0
Funding controls requirements definition	0	Requirements definition controls funding	1
Requirements are top down	1	Requirements are bottom up	2
Users/stakeholders are identified and interviewed (market survey)	1	No special consideration to identify users/stakeholders	0
Each requirement has a singular concept	3	Some requirements are compound statements	0
Requirements definition minimized when funding short	0	Program scope may reduce, but requirements definition completed	1
Requirements extraction has formal process	1	Requirements extraction ad hoc	0
Change procedures formal	1	Change procedures ad hoc	0
Users/stakeholders somehow involved in requirements definition	1	Program team only involved in requirements definition	0
Management sets requirements for developers	0	Developers at least partially involved in setting requirements	1
Requirements changed at least once since baseline established prior to new version	0	Requirements in baseline has not changed prior to new version or upgrade	1
No ranking of requirements	0	Requirements have priorities assigned	1
Use-case diagrams (or other models or scenario developments)	2	No models used for requirements extraction	0
Requirements changes informal	0	Requirements changes formal	1
Plan to "freeze" requirements as some designated milestone	1	No provision for "freezing" requirements	0
Requirements must be traceable	1	Origin of requirements not important	0
Requirements must be testable	3	System developed must be testable	1
Test plans to determine requirements implemented	2	No test plans needed for requirements verification	0
Requirements have priorities in implementation	1	All requirements must be implemented	0
Some requirements have multiple statements or ideas	0	One idea, one statement per requirement	2

Requirements Management (page 1 of 2) score □

Program Name______________

Date_______

ANSWER THIS BLOCK OF QUESTIONS ONLY IF A SEQUENTIAL OR WATERFALL APPROACH IS USED FOR DEVELOPMENT (Requirements page 2 of 2)			
Requirements first, then initial development work	1	Initial development work then requirements	0
Requirements documentation driving development	1	Requirements documentation developed in parallel/after development	0
User feedback considered during development	1	After development starts, user feedback serves as input to new work	0
Change management procedures used strictly	1	Change management procedures as guidance only	0
Design decisions prior to or in parallel to requirements development	0	Design decisions only after approved requirements stabilized	1
Requirements summarized what we have developed	0	Requirements are the blueprint for development	1
Length of time for requirements work greater than development work	2	Length of time for requirements work less than development work	0
Requirements have design detail	0	No design detail in requirements	1
Requirements creep to be avoided	1	Requirements creep o.k., but need to be controlled	0
Freeze requirements at some point	1	Requirements are fluid throughout development	0
Formal change procedure	1	Informal change procedure	0
Change management plan	2	No change management plan	0
Requirements ambiguity always present to some extent	0	Requirements ambiguity unacceptable at any level	2
Testing considered up front during requirements determination	2	Testing considered down the line during development	1
Requirements development team members different from implementation	0	Those working on requirements, work on implementation	1
Start implementation as early as possible to help define requirements	0	Requirements must be defined prior to any implementation work	2
ANSWER THIS BLOCK OF QUESTIONS ONLY IF A PROTOTYPING, THROWAWAY, SYNCHRONIZE & STABILIZE, OR OTHER STRATEGY USED			
Develop prototype, then determine requirements	1	Determine requirements prior to any development work	0
Requirements testing done after each iteration	1	No testing	0
Individual changes as necessary	1	Only block changes made	0
Development team decides on changes after iteration	0	Users involved with changes	1
Changes based on feedback only from user for correction of problems	1	Changes to upgrade system and correct problems	1
Funding controls changes and change procedures	1	Changes control funding	1
Requirements documentation finalized prior to development	0	Requirements fluid throughout development (only freeze at end)	2
Requirements test plans completed prior to development	1	Requirements test plans completed after development	0
Requirements first, then initial development work	0	Initial development work then requirements	1
Use development effort to learn more about requirements	2	Define all requirements prior to coding anything	0
Requirements ambiguity always present to some extent	1	Requirements ambiguity unacceptable at any level	0
Requirements have design detail	1	No design detail in requirements	1
User feedback considered during development	1	After development starts, user feedback serves as input to new work	0
Get something to users as soon as possible for evaluation	2	Make sure it is complete before releasing	0
Management dictates requirements	0	Developm't team visually represent requirements through rapid prototyping	1
New requirements allowed after initial requirements defined	1	New requirements not allowed	0

Requirements Management (pg 2 of 2) score □ **+ pg 1 score** □ **= TOTAL SCORE** □ **Enter on QMM scoresheet blk a.**

Program Name______________

Date_______

Pair choice section TWO: (Estimation/Planning Management) choose most applicable term of the two for each row (page 1 of 2):

At least one estimation method used in program	1	No estimates	0
Formal derivation of product metric for estimation of size	1	Ad hoc size estimation	0
Ad hoc process evaluation	0	Formal derivation of at least one process metric	1
Develop work breakdown structure (WBS)	1	Assign work as needs arise	0
Estimates are developed to fulfill a data call only	0	Use estimates to plan program	1
Use estimates to sell program only	0	Estimates are useful to the project team for planning purposes	1
Resource evaluations made for program	1	No resource evaluation for planning	0
Use both bottom up & top down for estimate, use one stakeholders like	0	Use both bottom up & top down and evaluate significant differences	1
Estimates made and not updated	0	Estimates updated throughout program	1
Resource estimations used to adjust product size estimate	1	Estimations made irregardless of resources available	0
Estimations made to fit budget	0	Budget made from estimations	1
Estimations compromised to get program	0	Rather risk loss of program than compromise confident estimations	1
Cycle time estimations	1	No cycle time estimations	0
Event count estimations	1	No event count estimations	0
Lines of code (LOC) estimation	1	No LOC estimation	0
Function Point (FP) estimation	1	No FP estimation	0
Estimates by algorithmic methods	1	Estimates by analogy	1
Expert judgement for estimation	1	Ad hoc estimates	0
Estimates by algorithmic methods	1	Ad hoc estimates	0
Expert judgement for estimates	0	Estimates by analogy	1
Ad hoc estimates	0	Estimates by analogy	1
Bottom up estimates	1	Expert judgement	0
Top down estimates	1	Expert judgement	0
Ad hoc estimates	0	Any other estimate process	1
Fuzzy logic estimating method	1	No formal estimation methodology	0
WBS development from estimates	1	WBS development in parallel or prior to estimation completion	0
Critical path of program determined	1	Tasks developed but no path is identified	0
Estimators are program team members	1	Estimators are outside program team	0
Management only on estimations	0	All team members involved in estimation process	1
Estimates updated at reviews	1	No updates of estimates	0
Estimates updated at reviews	0	Estimates constantly updates (in between reviews, too)	1
Estimate procedures stay the same	1	Estimate procedures change	0
Stakeholders are part of estimation process	1	Stakeholders brief estimations after completion	0
Estimates are used beyond initial selling of program	1	Estimates are one time events, used for a specific purpose once	0
WBS has objective measure of completeness	1	Important to have WBS as guide, not rigid implementation	0

Estimation/Planning Management page 1 of 2 score □

Program Name______________

Date_______

Pair choice section TWO: (Estimation/Planning Management) choose most applicable term of the two for each row (page 2 of 2):

Life cycle estimates	1	Estimates for program initiation only	0
System upgrades (SCR) software change requests estimated individually	1	Systems upgrades estimated as whole	0
Estimates for on-going resources needed to maintain s/w	1	Estimates for maintenance not done	0
Informal re-estimates during development	0	Formal re-estimates at pre-defined milestones	1
Formal re-estimates when amendment changing the system is introduced	1	Informal re-estimates when amendment changing the system	0
person in-charge of estimation walks in a managers office to get an opinion	0	Meeting(s) organized for purpose of performing cost estimations	1
Factor analysis prior to commencement of program	1	None done	0
Change control procedures set in place	1	No set procedures	0
Elapsed time and actual work time estimates	1	one or the other or neither	0
No schedule created	0	Schedule created	1
Schedule not updated	0	Schedule updated	1
Schedule followed	1	Schedule not followed	0
Tasks identification arises as program progresses	0	Detailed level tasks identified prior to program initiation	1
Scope of program understood by all	1	Scope not explicitly defined	0
Quality factors and criteria identified	1	No explicit quality factors defined	0
No project tracking tools used	0	Project tracking tools used	1
CSCIs identified and tasked	1	CSCIs not explicitly identified	0
Expectations are managed via estimations	1	Estimations are made to fit preconceived expectations	0
No cost schedule developed	0	Cost schedule developed	1
No resource schedule developed	0	Resource schedule developed	1
Team members, management know at any time if in budget & schedule	1	Exact budget & schedule status somewhat unclear to at least some	0
Individual program phases are estimated	1	Only top level program estimated	0
Stakeholders/users emphasis understood- quick to field or all complete	1	Program management sets delivery tradeoffs without outside input	0
Testing planned with initial program planning	1	Testing no in initial planning	0
Documentation not considered in initial planning	0	Documentation part of initial planning	1
Hardware considered in estimations	1	Software only considered	0
No formal schedule/cost tracking	0	Formal procedures established for tracking cost and schedule	1
Earned value set up	1	Earned value not used	0
Estimations omit documentation planning	0	Documentation in estimates	1
Training omitted in estimates	0	Training part of estimates	1
Earned value set up, but not tracked	0	Earned value tracked	1
Detailed planning done with incomplete set of requirements	0	Detailed planning done with detailed set of requirements	1
Complete infrastructure support mechanism understood for estimations	1	No consideration of infrastructure done for estimations	0
Team possibilities considered for planning of program	1	No consideration for outside teaming possibilities	0
Work Breakdown Structure (WBS) set up	1	No WBS completed	0

Estimation/Planning Management pg 2 of 2 score □ **+ pg 1 of 2 score** □ **= TOTAL SCORE** □ **Enter QMM scoresheet blk b.**

Program Name______________

Date_______

Pair choice section THREE: (People Management) choose most applicable term of the two for each row (page 1 of 2):

Human Resources

Program team members have clearly defined, segmented roles	0	Work responsibilities are shared	1
Formal team building procedures are used	1	No formal team building emphasized	0
Program manager flexible regarding work hours	1	Program manager maintains strict standards for work hours	0
Big picture conveyed to all team members by program management	1	Program management focuses on the partitioned tasks with team	0
People issues dealt with primarily through indirect methods (email, memo, etc)	0	People issues dealt with primarily through direct methods (face-to-face)	1
Training is required and planned on a regular basis	1	Training is ad hoc	0
Each team member is educated on and understands overall program and their roles	1	Team members only know their respective areas	0
Consideration for team members' career goals are reflected in assignments	1	Team members must adapt to tasks that are assigned	0
Team members assignments and responsibilities are mostly dictated by PM	0	Assignments and responsibilities are discussed and agreed upon with PM	1
Management leads in problem solving	0	Management facilitates and lets team lead in problem solving	1
Management welcomes problems as challenges and opportunities	1	Management views problems as obstacles and grounds for punishment	0
Team members participate in performance evaluations of peers	1	Personnel evaluations are strictly PM responsibility	0
Management reinforcement feedback sparse and inconsistent, if any	0	Management provides timely reinforcement feedback for positive behaviors	1
Management provides basic needs of office facilities fairly well	1	Office facilities are a drawback to working in the program	0
Working conditions are fairly comfortable, time off policy "flexible"	1	Working conditions and time off policy is inconsistent and difficult at times	0

Communication:

Communications primarily written (email, memo, etc.)	1	Communications primarily verbal (face-to-face)	1
Detailed instructions: oral presentation, follow-up email, memo, etc.	1	Email, memo, etc. only	0
Formal communication protocol	1	Informal communications	0
External vertical communications restricted	0	External vertical communication allowed	1
Coders notebook, weekly accomplishment reports required	1	Not required	0
User-coder relationship established, encouraged, and mediated	1	User-coder interaction minimized	0
Meetings structured to minimize wasted time	1	Meetings unstructured and open ended	0
Meetings have agenda, objectives, and conclude with action items	1	Meeting agenda fluid and open ended	0
Program management and coder communication face to face	1	Program management and coder communication primarily email	0
Program team updated regularly regarding organizational & program status	1	Meetings infrequently scheduled	0
Open communications is encouraged	1	Communication through chain of command only is encouraged	0
Program manager accessible for discussions	1	Program manager difficult to get an appointment to see	0
Program management (PM) is viewed as separate from team	0	PM mixes with team frequently	1
Management regularly holds team meetings	1	Meetings are sporadic	0
Meetings are structured with definite goals and objectives	1	Meetings are informal	0
Program management is generally easy to reach and talk to	1	PM is usually hard to get a hold of and difficult to talk to	0
Team-program manager relationship adult-adult	1	Team-program management relationship parent-child	0
Schedules are spontaneous and poorly communicated	0	Schedules must be fixed and rigidly followed and formally reported	1
work is seen as complex processes involving team working together	1	Work broken into pieces with minimal team member interaction	0
Action items sometimes are not followed through	0	Action items communicated and followed through thoroughly	1
Team members require frequent clarifications by PM for assigned tasks	0	Team members rarely require clarifications by PM for assigned tasks	1

Program Name______________

Date_______

Pair choice section THREE: (People Management) choose most applicable term of the two for each row (page 2 of 2):

Leadership:

Long range organizational vision	1	Short term program and immediate work focus	0
Lead through personal attention to others	1	Action-oriented leadership approach	1
Run as much of the organization as possible	0	Let team make decisions as much as possible	1
Direct and domineering style	0	Encourage independence in others	1
Traditional leaders respect hierarchy	0	Do what needs to be done	1
Win cooperation rather than demand it	1	Tough-minded with others	0
Act strongly and forcefully in the field of ideas	0	Prefer to lead other independent types while seeking autonomy for self	1
Consults with team members to find solutions to problems	1	Consults team members to get validation of PM's predetermined solutions	0
Keep people well informed	1	Only as much knowledge as necessary for their work	0
Make things happen by focusing on the immediate problems	1	Long range focus and de-emphasize current problems	1
Manage others loosely and prefer minimal supervision	1	Follow traditional procedures and rules conscientiously	0
Leadership, management decisions exclusively by program management	0	Program management makes decisions but gets inputs from team	1
Team-program manager relationship adult-adult	1	Team-program management relationship parent-child	0
Program management makes decisions but gets inputs from team	0	All program team members responsible for program decisions	1
When a problem arises: management takes over to solve it	0	Management lets the team solve the problems	1
Leadership is do as I say, not do as I do	0	Leadership by example	1
Program expectation not influenced by PM	0	Program expectation managed by PM	1
PM gives freedom to team, but does has no mentoring for members	0	PM empowers teams by mentoring members to be leaders	1
Program management waits and sees what happens then plans	0	Management plans far in advance	1
Program management is reacts to emergencies	0	Management is one step ahead of problems	1
Facilitative approach to solving problems	1	Take charge readily and often	0
Program management is complex, takes much time to understand	0	Management is simple, easy to figure out	1
Program management prefers to plunge right in	0	Takes time to separate things to be done and order of doing them	1
Program management reacts to needs of the moment	0	Methodically follows plans	1

Technical Competency of the program manager:

PM has technical experience particular to the particular s/w program	1	PM relies on team members solely	0
PM participates in technical reviews	1	PM only in non-technical reviews	0
PM participates in making technical decisions when problems arise	1	PM delegates technical questions	0
PM does not get involved discussing technical options	0	PM contributes to technical options being discussed	1
PM does not review technical options and decisions	0	PM reviews technical options and decisions	1
PM actively attempts to keep up-to-date with current technology and standards	1	PM is removed from cutting edge technology issues	0
PM receives technical periodicals and occasionally references applicable articles	1	PM doesn't read periodicals nor references current articles to team	0
PM doesn't have technical background (or education)	0	PM has technical background (or education)	1
Team members avoid PM when they need technical advice	0	Team members generally consider talking to PM regarding technical issues	1

HR □ **+ Comm.** □ **+ Leadership** □ **+ Tech. Competency** □ **= People Mgmt. score** □ **Enter on blk c.**

Program Name______________

Date_______

Pair choice section Four: (Risk Management (RM)) choose most applicable term of the two for each row (page 1 of 2):

RM is formal and documented	1	RM is informal, if at all	0
A risk management plan exists	1	No risk management plan is developed	0
RM is more of a data call than a useful document	0	RM drives decisions on the program	1
RM is done prior to the program beginning	0	RM is done prior and during program execution	1
RM is only done during the program execution	0	RM is done prior and during program execution	1
Risks are generalized through the whole program	0	Risks are categorized	1
Risk management is done internally, only	0	An outside organization also contributes to the RM process	1
Risk is a management function	0	Risk is a program team function	1
Risk are precisely articulated	1	Risks are generalized, if at all	0
Each risk has a consequence	1	Consequences are generalized, if at all	0
A mitigation strategy is completed for each risk	1	Mitigation strategy is generalized, if at all	0
Contingency plans are developed for a RM plan	1	Contingency plans are ad hoc as problems arise in the program	0
Risks are anticipated	1	If problems arise, management will deal with it	0
The program doesn't have any risk	0	Programs that do not have risk, have problems	1
Risk management is automated	0	Risk management may use tools, but depend on human input	1
Risks are assigned probabilities	1	Probabilities are not relevant for RM	0
All risks are potential problems, relative priorities for risks are not useful	0	Risks are weighed relative to other program risks and thus prioritized	1
Risk management information is only shared internally	0	Risk management information is shared with all stakeholders	1
Risk analysis uses ordinal rankings	0	Risk analysis uses actual measurements with a mathematical model	1
Regret analysis used	1	No regret analysis done	0
Attach probabilities to future events	1	No probabilities associated with future events	0
Assessing risks with mechanical methods	0	Risks should be compared to other risks and sorted	1
Risk status tracked	1	Not tracked	0
Technical risks examined	1	No technical risks examined	0
Process risks examined	1	No process risks examined	0
Product risks examined	1	No product risks examined	0
Stakeholder/user risks examined	1	No examination of stakeholder/user risks	0
Checklists used to identify risks	1	No checklists used	0
Risks are tracked	1	No tracking or monitoring of risks	0
Each risk has an impact	1	No impact analysis of risk	0
Each risk has a mitigation plan	1	No individual risk mitigation	0
Risks monitored by priority	1	No special attention to track higher priority risks	0
Risk assessment is formalized	1	No formal risk assessment	0
Risk control is formalized	1	No formal risk control	0
Integration risks not considered	0	Integration risks examined	1

Risk Management page 1 of 2 score □

Program Name______________

Date_______

Pair choice section Four: (Risk Management (RM)) choose most applicable term of the two for each row (page 2 of 2):

Risks to cost	1	No cost risks examined	0
Unforeseen risks have occurred in program	0	Any risk that came up had been identified previously	1
Personnel risks examined	1	No personnel risks examined	0
Estimation risks examined	1	No estimation risks examined	0
Planning risks examined	1	No planning risks examined	0
Requirements risks examined	1	No requirements risks examined	0
Resource risks examined	1	No resource risks examined	0
Risk management plan updated regularly	1	No regular risk management plan updates	0
Risks charted	1	Risks not charted	0
Performance risks examined	1	Performance risks not examined	0
Program management self risks examined	1	No program management risks examined	0
Risk from program constraints examined	1	No program constraint risks examined	0
Each category of risks are prioritized	1	No prioritization	0
Each category of risks are evaluated for impact	1	No impact analysis performed	0
Each category of risks have control strategy	1	No control strategy	0
Documentation risks examined	1	No documentation risks examined	0
Regret matrix tracked	1	No regret matrix or not tracked	0
Communication of risk activities are facilitated	1	No facilitation or promotion of communication of risk activities	0
Taxonomy-based questionnaire used to identify risks	1	Taxonomy-based questionnaire not used	0
Associated hardware risks examined	1	No consideration for hardware risks	0
Integration risks examined	1	Integration risks not examined	0
Communication risks examined	1	Communication risks not examined	0
Leadership risks examined	1	Leadership risks not considered	0
Risk avoidance considered for certain risks	1	Risk avoidance not considered for risks	0
Risk documentation forms used	1	No risk documentation forms used	0
Dependency risks examined	1	No dependency risks examined	0
Alternatives like risk avoidance considered for high risk items	1	No consideration of risk avoidance	0
Documented risk statements use a condition-consequence type format	1	Condition-consequence of risk statements not clearly defined	0
No assignment of ownership of risk mitigation action	0	Each risk mitigation action is assigned to an individual for resolution	1
Calculation of risk exposure made (probability X loss, for each risk)	1	No risk exposure calculations	0
Oral communication of risks only	0	Risks written in a way that communicates nature and status of factors	1
Triggers used to quantify risk conditions present	1	Risk conditions present are all subjective	0
Risk "czar" in program for monitoring risks	1	No special positions/responsibilities for risk monitoring	0
Post-program review completed (scheduled) for unanticipated problems ID	1	No post-program reviews completed or scheduled	0
No schedule risks examined	0	Risks to schedule investigated	1

Risk Management pg 2 of 2 score □ **+ pg 1 of 2 score** □ **= TOTAL SCORE** □ **Enter on QMM scoresheet blk d.**

Program Name______________

Date_______

Program Name________ **YES-NO-N/A Questionnaire Scoring Template** Date______

No.	Requirements Management Questionnaire	Yes	No	N/A
1	PM chose to have a formal requirements list	1	0	0
2	Requirements recorded in some way	2	-1	0
3	Written requirements were part of some formal document	1	0	0
4	Written requirements were informal	1	2	0
5	At least some requirements were oral only	-2	1	0
6	All stakeholders were identified	2	-1	0
7	All stakeholders participated in the requirements extraction	2	0	0
8	Some stakeholders participated in the requirements extraction	1	0	0
9	Management extracted requirements, no stakeholder involvement	1	2	1
10	Management passed requirements to development team	1	0	0
11	Stakeholders not involved in Management extraction, but approves	-1	0	0
12	Management gets inputs from stakeholders, then develops requirments	1	0	1
13	Developers work informally with users to arrive at requirements	1	0	0
14	Same as 13, but management oversees and formalizes	2	0	0
	If a waterfall or sequential development strategy:			
15	All requirements complete before design	1	-3	0
16	Some requirements left incomplete prior to design	-1	0	0
17	Requirements informal prior to design effort	-1	0	0
18	Requirements serve as input	1	-1	0
19	Length of time for requirements work greater than development work	2	-1	0
20	Requirements developed in parallel to design	-1	1	0
OR	***If a prototype, throwaway, or other development strategy:***			
15	Learn about requirements through development efforts	1	-1	0
16	No coding until all requirements are defined	-3	1	0
17	Requirements formal prior to design effort	-1	0	0
18	Requirements serve as output	1	-1	0
19	Requirements definition work in parallel to development efforts	2	-1	0
20	Requirements developed in parallel to design	1	-1	0
21	Are requirements frozen at some phase	1	-1	0
22	Change management exists	3	-3	0
23	Change management is formal	1	0	0
24	Project strategy is consistent throughout development	1	0	0
25	Requirements are updated	1	0	0
26	Configuration Management (CM) exists	3	-3	0
27	CM is formal	1	0	0
28	Requirements are testable	2	-2	0
29	Requirements testing considered/implemented during extraction	2	0	0
30	Requirements testing plan exists	2	0	0
31	Requirements testing is formal	1	0	0
32	All requirements have priorities	2	-2	0
33	All requirements must be implemented	0	1	0
34	Requirements are tested	1	-1	0
35	All requirements are equally important	0	1	0
36	At least some requirements have priorities	1	0	0
37	All requirements are traceable	1	0	0
38	Traceability not important	0	1	0
39	Each requirement has an author	1	0	0
40	Who authored requirement is not important	0	1	0
41	Initial set of requirements to be implemented, no requirements creep	0	1	0
42	Structured and tracked changes to requirements only	1	-1	0
43	Change is inevitable, changes allowed at all times	-1	1	0
44	Change is inevitable, but changes limited	1	0	0
45	Requirements control funding	1	0	0
46	Requirements history kept	1	-1	0
47	Baseline established for requirements at some point prior to develop	2	-2	0
	TOTAL SCORING			

Enter total score on QMM score sheet block e.

Program Name________ **YES-NO-N/A Questionnaire Scoring Template** Date_______

No.	Estimation/Planning Questionnaire	Yes	No	N/A	
1	A volume product metric used (LOC, # of files, # of screens, pages of doc)	1	0	0	
2	Measure used for various product elements (modules, components, CSCI)	1	0	0	
3	Product measures made by phase (amt at implementation, LOC changed at unit test)	1	0	0	
4	Other product attributes measured (FP, throughput, mem cap, cyclomatic complexity)	1	0	0	
5	Product metrics tracked and updated throughout program execution	2	-1	0	
6	Event count process metric used (# defects in test, reqmt changes, milestones met)	1	0	0	
7	Time measure process metric used (cycle time)	1	0	0	
8	Process metrics tracked and updated throughout program execution	2	-1	0	
9	Program cost estimations made from product or process metrics	1	0	0	
10	Program cost estimations tracked and updated to reflect progress/changes	1	0	0	
11	Factor analysis performed on program	1	0	0	
12	Program's primary purpose, including major functions and deliverables known	2	-1	0	
13	Work breakdown structure developed	2	-1	0	
14	Task estimated with realistic expectations of productivity probabilities	1	-1	0	
15	Schedules developed based on realistic expectations	1	-1	0	
16	Schedules tracked and updated based on new information	1	-1	0	
17	Detailed activity lists used for clearly defined completed/not completed tasks	1	-1	0	
18	Quality assurance plan or similar to aid in detecting defects early in program	1	-1	0	
19	COCOMO estimates performed	1	-1	0	
20	CSCI clearly defined and tasked	2	-1	0	
21	Estimates completed ad hoc	-2	0	0	
22	Gantt charts used and updated	1	-1	0	
23	Resource estimations (working hrs, job categories, task activities) done	1	-1	0	
24	Earned value established	2	-1	0	
25	Earned value tracked throughout program	2	0	0	
26	Quality expectations established for product with users and stakeholders	1	-1	0	
27	Critical path for program tasks developed and tracked	2	-1	0	
28	Meaure of effectiveness (MOE) or Figure of merit established and tracked	1	0	0	
29	Estimates are updated routinely	2	-1	0	
30	Schedules are updated routinely	2	-1	0	
31	Estimations are made by program management (top-down)	1	0	0	
32	Estimations are made by program team members (bottom-up)	2	0	0	
33	Automated program tracking used	1	0	0	
34	PM usually thorough in tracking and reporting schedules and financials	1	-1	0	
35	WBS developed only as data call, not used in planning	-1	0	0	
36	Earned value used to track program progress	2	-1	0	
37	PM insists on prioritizing work reduction as schedule/funding compromised by stakeholders	1	-1	0	
38	Estimations are done using both top down and bottoms up approaches	2	-1	0	
39	All program team members involved in planning process	2	-1	0	
40	Hardware also considered in estimation process	1	-1	0	
41	Program history compiled	1	0	0	
42	System upgrades (SCR) software changes requests estimated individually	1	-1	0	
43	Management duties apart of each team member's responsibilities	-1	1	0	
44	PM dictates schedules to program team	-1	0	0	
45	Code reviews planned in schedule	1	-1	0	
46	Defined tangible milestones established for program tasks	2	-1	0	
47	Test planning done at the start of the program	1	-1	0	
48	Estimations are completed by those performing the tasks	1	-1	0	
49	Sensitivity analysis performed for program choices	1	-1	0	
50	Software deployment planning completed	1	-1	0	
	TOTAL SCORING				

Enter total score on QMM score sheet block f.

Program Name________ **YES-NO-N/A Questionnaire Scoring Template** Date______

No.	People Management Questionnaire	Yes	No	N/A	
1	PM is accessable in person by each team member	1	0	0	
2	PM is accessable via email by each team member	1	0	0	
3	PM is accessable via phone by each team member	1	0	0	
4	PM not only considers a person's suitability, not also desire to be on a team	1	0	0	
5	PM consults with each team member regarding their career goals	1	0	0	
6	PM regularly holds meetings to inform team of program progress	2	-1	0	
7	PM solicits opinions from team members before making decisions	2	-1	0	
8	PM lets teams make decisions affecting their work	1	0	0	
9	PM frequently makes decisions without any consultation with members	-2	2	0	
10	PM understands the technology/language of the program	1	0	0	
11	PM is able to communicate with other the technical issues in the program	1	-1	0	
12	PM prioritizes problems or conflicts within the program	1	0	0	
13	PM assists team members in developing/advising of career path	1	-1	0	
14	PM empowers program members to recommend hiring new team members	1	-1	0	
15	PM empowers program members to recommend firings of other members	1	-1	0	
16	PM specifically assigns work to each program member	1	-1	0	
17	PM sets communication protocol to be followed	1	0	0	
18	PM allows unrestricted communications	1	0	0	
19	PM readily makes tough decisions	1	-1	0	
20	PM takes control in difficult/ problem areas	1	0	0	
21	PM looks ahead to new programs, new upgrades of existing program	1	0	0	
22	PM maintains regular communications with all stakeholders	2	-1	0	
23	PM maintains regular communications with users	2	-1	0	
24	PM encourages program team communication with users	1	-1	0	
25	PM encourages program team communication with stakeholders	1	-1	0	
26	PM facilitates horizontal communication within program	1	-1	0	
27	PM facilitates communication during integration	1	-1	0	
28	PM holds meetings without clear objectives listed prior to meeting	-1	2	0	
29	PM must approve all decisions within the program	-1	1	0	
30	PM must approve all interactions with stakeholders	-1	1	0	
31	PM must approve all interactions with users	-1	1	0	
32	PM makes all presentations to stakeholders/users	0	1	0	
33	PM is considered "flexible" in terms of program members personal issues	1	0	0	
34	PM, at least occasionally, schedules/promotes outside work team activities	1	0	0	
35	PM is readily willing to listen to program problems and complaints	1	-1	0	
36	PM takes action to resolve program problems and complaints	1	-1	0	
37	PM is generally respected by stakeholders, users, and organization	1	-1	0	
38	PM sometimes fails to grasp important technical issues in program	-1	1	0	
39	PM recruits program team members from outside organization	1	-1	0	
40	PM directs what needs to be done and directs how to do it	-1	1	0	
41	Program personnel have clearly defined specific tasks	0	1	0	
42	Although individual's tasks are specific, each exposed to the "bigger picture"	2	-1	0	
43	PM has clearly defined his/her expectations for each individual	2	-1	0	
44	PM delegation of duties is usually seemless in execution	1	0	0	
45	PM acts as facilitator to solving personnel conflicts	2	-1	0	
46	PM attempts to motivate individuals on the program team	2	-1	0	
47	PM clearly separates technical from managerial roles for individuals	0	1	0	
48	PM directs how he/she expects the task to be accomplished	0	1	0	
49	PM directs what needs to be done, but does not direct how	2	-1	0	
50	PM attempts to spotlight individuals in the program for positive exposure	2	-1	0	
	TOTAL SCORING				

Enter total score on QMM score sheet block g.

Program Name________ **YES-NO-N/A Questionnaire Scoring Template** Date______

No.	Risk Management Questionnaire	Yes	No	N/A	
1	Risk Management (RM) is specifically an activity in the program	4	-4	0	
2	RM is formal and documented	3	-3	0	
3	A specific RM plan exists	2	-2	0	
4	RM is required in the program, but not used during the program	-1	1	0	
5	RM is done prior to the program execution	1	0	0	
6	RM is done by an outside entity to the development	1	0	0	
7	RM is done internally only	0	1	0	
8	RM is both internally performed and externally assessed	1	-1	0	
9	RM planning occurs during or after major milestones in the program	1	-1	0	
10	Risk Assessment is only a management function	0	1	0	
11	RM is informal or non existent	-1	1	0	
12	There is a RM plan, but it is not updated or tracked	1	0	0	
13	Risks are only generalized	-1	0	0	
14	Each risk is delineated	1	0	0	
15	Each risk has a consequence	1	0	0	
16	Each risk has a likelihood rating of some sort	1	0	0	
17	Each risk has a mitigation strategy	1	0	0	
18	Risk Management is automated	1	0	0	
19	Risks are tracked	2	-2	0	
21	Regret analysis performed	2	0	0	
22	RM drives decisions in the program	3	-2	0	
23	Risks have probabilities	1	0	0	
24	Risk Management is ad hoc	-3	0	0	
25	RM information is shared with all stakeholders (as appropriate)	1	0	0	
26	Risks are weighed relative to other program risks	1	0	0	
27	Risk Assessment is a program team activity	1	0	0	
28	Risk Assessment done prior to program start	2	-1	0	
29	Risk Assessment includes personnel risk	1	-1	0	
30	RM uses tools, but depends on human decisions	2	-1	0	
31	Risk Assessment includes cost risks	1	0	0	
32	Risk Assessment includes schedule risks	1	0	0	
33	Risk Assessment includes technology risks	1	-1	0	
34	Risk Assessment is briefed organization structure above program manager	1	-1	0	
35	Risk Assessment includes requirements risks	1	-1	0	
36	Risk Assessment includes user risks (too little involvement of user)	1	0	0	
37	Risk Assessment includes documentation risks	1	0	0	
38	Risk Assessment includes integration risks	1	-1	0	
39	Risk Assessment includes interface risks (non-standard)	1	-1	0	
40	Risk Assessment includes continuing requirements change (feature creep)	1	-1	0	
41	Risk Assessment includes dependent projects/programs risks	1	0	0	
42	Documentation proof exists to demonstrate following risk management plan	1	0	0	
43	High risk have measured tracking (high profile status)	1	0	0	
44	Organizational history used to search for risks	1	0	0	
45	Other organizational checklists used for risk assessment	1	0	0	
46	Internal organizational checklists used for risk assessment	1	0	0	
47	Risk Assessment information contributed to internal or other database	1	0	0	
48	Risk Assessment includes internal organization risks	1	0	0	
49	Risk Assessment includes stakeholder risks	2	-1	0	
50	No risk management needed; program is straightforwarded & understood	-3	3	0	
	TOTAL SCORING				

Enter total score on QMM score sheet block h.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Boehm, B.W., *Software Engineering Economics*, Prentice-Hall, Inc., 1981.
2. Thielsen, D., "The Commando Returns", *Software Development*, March, 1999.
3. Bahill, A.T. and Dean, F., *Discovering System Requirements*, Bahill and Dean, 1997.
4. Humphrey, W.S., *A Discipline for Software Engineering*, Addison-Wesley Publishing Co., 1995.
5. Davis, A.M. and Leffingwell, D.A., "Making Requirements Management Work for You", *Crosstalk: The Journal of Defense Software Engineering*, April, 1999.
6. Launi, J.D., "Creating a Project Plan", *Software Development*, May, 1999.
7. Wiegers, K., "Know Your Enemy: Software Risk Management", *Software Development*, October, 1998.
8. Ambler, S., "Engineering Object-Oriented Requirements", *Software Development*, March, 1999.
9. Wiegers, K., "Writing Quality Requirements", *Software Development*, May, 1999.
10. Wiegers, K., "Automating Requirements Management", *Software Development*, July, 1999.
11. IEEE Software, "Software's Ten Essentials", *Best Practices, Vol. 14, No. 2*, March/April, 1997.
12. Machniak, M., "Interview with CAPT (ret.) L. Preston Brooks Jr., Software Program Manager for the Advanced Information Technology Group at SAIC. Previous Eldyne software program manager for the Financial Information Support System/ Expenditure Tracking System (FISS/ETS) development program", 3 September, 1999.
13. Machniak, M., "Interview with CAPT (ret.) Gerald Nifontoff, Advanced Programs Manager for Lockheed Martin Undersea Systems. Previous Navy program manager for the Surveillance Towed Array Sensor (SURTASS) program", 7-8 September, 1999.
14. Boehm, B., "Making RAD Work for Your Project", *IEEE Computer Society*, March, 1999.
15. Phillips, D., "Throwaway Software", *Software Development*, October, 1999.
16. Cusumano, M. and Yoffie, D., "Software Development on Internet Time", *IEEE Computer Society,* October, 1999.
17. Wiegers, K., "First Things First: Prioritizing Requirements", *Software Development*, September, 1999.
18. Heberling, J., "Software Change Management", *Software Development*, July 1999.
19. Osmundson, J., "Class Notes from Naval Postgraduate School Course IS4300, Software Project Management", 1998.
20. Weber, R., "Time Management Essentials", *Seminar Series,*1998.
21. Fairley, R., *Software Engineering Concepts,* McGraw-Hill, Inc., 1985
22. Pressman, R., *A Managers Guide To Software Engineering*, McGraw-Hill, Inc., 1993.

23. Brooks, Jr., F.P., *The Mythical Man-Month,* Addison Wesley Longman, Inc., 1995.
24. Strauss, J., "The Softer Side of Project Management", *Crosstalk-The Journal of Defense Software Engineering,* July 1998.
25. Pfeffer, J. and Veiga, J., "Putting People First for Organizational Success", *IEEE Engineering Management Review*, Fall 1999.
26. Luthans, F. and Stajkovic, A., "Reinforce for Performance: The Need to Go Beyond Pay and Even Rewards", *IEEE Engineering Management Review*, Fall 1999.
27. Galorath, D., Fischman, L., and McRitchie, K., "Driving Quality Through Parametrics", *Crosstalk-The Journal of Defense Software Engineering,* November 1998.
28. Wedberg, G., "Pro-Active Metrics", *Crosstalk-The Journal of Defense Software Engineering,* August 1998.
29. Curtis, B., Hefley, W., Miller, S., "Overview of the People Capability Maturity Model" (CMU/SEI-95-MM-01), *Software Engineering Institute, Carnegie Mellon University*, September 1995.
30. Curtis, B., "Which Comes First, the Organization or Its Processes?", *IEEE Software*, November/December 1998.
31. Schaaf, R. J., "Whence Leadership?", *IEEE Software,* March/April 1999.
32. Burgess, P. and Street, J., "Learning to Lead", *JP Training and Development Associates*, July 1999.
33. Briggs-Myers, I., *Introduction to Type*, Consulting Psychologists Press, Inc., 1993.
34. Machniak, M., "Interview with Julie Streets", *Partner, JP Training and Development Associates,* August 3, 1999.
35. Pickering, J., *Building High-Performance Organizations for the Twenty-First Century,* Commonwealth Center for High-Performance Organizations, Inc. and The Federal Executive Institute, 1999.
36. Machniak, M., "Software Program Management Focus Group #1", *SPAWAR Systems Center-San Diego (SSC-SD),* October 15, 1999.
37. Loomis, M., "Managing Collaborations", *Software Development,* January 1998.
38. Machniak, M., Interview with Dr. John Pickering, Organizational Consultant, September, 1999
39. Keuffel, W., "Planning for and Mitigating Risk", *Software Development,* September 1999.
40. Higuera, R. and Haimes, Y., "Software Risk Management", *Software Engineering Institute,* June 1996.
41. Machniak, M., "Software Program Management Focus Group #2", *SPAWAR Systems Center-San Diego (SSC-SD),* October 26, 1999.
42. Meindl, J. R., and Ehrlich, S. B., "The Romance of Leadership and the Evaluation of Organizational Performance", *Academy of Management Journal,* 1987.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
8725 John J. Kingman Rd., STE 0944
Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library 2
Naval Postgraduate School
411 Dyer Road
Monterey, California 93943-5101

3. Chairman, Code CS 1
Naval Postgraduate School
Monterey, California 93943-5100

4. Dr. Luqi, CS/Lq 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5100

5. Dr. Man-Tak Shing, Code CS/SH 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5100

6. Dr. Valdis Berzins, Code CS/BE 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5100

7. Dr. J. Bret Michael, Code CS/MJ 1
Computer Science Department
Naval Postgraduate School
Monterey, California 93943-5100

8. Dr. John Osmundson, Code AG/JO 1
C3 Academic Group
Naval Postgraduate School
Monterey, California 93943-5100

9. Software Engineering Process Office, Code D12 1
SPAWAR Systems Center
53560 Hull Street
San Diego, California 92152-5100

10. Navigation and Applied Sciences Department, Code D30 1
SPAWAR Systems Center
53560 Hull Street
San Diego, California 92152-5100

11. Gerald Nifontoff.. 1
1777 Puterbaugh Street
San Diego, California 92103

12. L. Preston Brooks, Jr.. 1
11744 Caminito de las Missiones
San Diego, California 92128

13. Martin J. Machniak. 1
160 Shadow Glen Court
El Cajon, California 92019